

Entwicklung neuer Diagnoseverfahren auf der Basis multivariater Methoden und deren Anwendung in einem Softwareinformationssystem zur Anlagenüberwachung

Von der Fakultät für Ingenieurwissenschaften der
Abteilung Elektrotechnik und Informationstechnik
der Universität Duisburg-Essen

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

genehmigte Dissertation

von

Peter Engel

aus

Altdöbern

1. Gutachter: Prof. Dr.-Ing. Steven X. Ding
 2. Gutachter: Prof. Dr.-Ing. Gerhard Lappus
- Tag der mündlichen Prüfung: 25.04.2012

Vorwort

Die vorliegende Arbeit entstand während meiner beruflichen Tätigkeit als Teamleiter Consulting und Verantwortlicher für die Produktentwicklung des Asset-Management-Systems zedas®asset der Firma PC-Soft GmbH.

Herrn Prof. Dr.-Ing. Steven X. Ding danke ich für die wissenschaftliche Betreuung der Arbeit sowie für die Hilfe und Förderung. Bei Herrn Prof. Dr.-Ing. Lappus bedanke ich mich freundlich für seine Unterstützung und das entgegengebrachte Interesse.

Mein Dank gilt Herrn Dipl.-Ing. Wolfgang Jahn, der mir in seiner Funktion als Geschäftsführer der Firma PC-Soft GmbH diese Arbeit ermöglicht hat.

Ich bedanke mich ausdrücklich bei Herrn Prof. Dr.-Ing. Torsten Jeinsch für seine Unterstützung, seine Anregungen und Diskussionen sowie für die Schaffung der Arbeitsmöglichkeiten. Des Weiteren möchte ich mich bei den Kollegen des Fachbereiches Automatisierungstechnik der Universität Duisburg-Essen und der Hochschule Lausitz für ihre kollegiale Zusammenarbeit bedanken. Speziell bin ich den Herren Dipl.-Ing. (FH) Daniel Schoch, Dipl.-Ing. (FH) Torsten Stargala und Dipl.-Ing. (FH) Gunar Gruss zu Dank verpflichtet.

Der Firma Förderanlagenbau Magdeburg GmbH und besonders Herrn Dipl.-Ing. Bernd Petermann und Dr.-Ing. Sergej Kaverinski danke ich für die Unterstützung und für die Bereitstellung von Messdaten.

Entschuldigen möchte ich mich bei meiner Familie, die in dieser Zeit sicherlich etwas zu kurz gekommen ist, mir aber sehr viel Verständnis entgegen brachte sowie beim Schwimmverein Senftenberg und bei Freunden und Bekannten, welche ich in dieser Zeit vernachlässigte.

Senftenberg, im November 2011

Peter Engel

Für meine Eltern, meine Ehefrau Katrin sowie für meine Kinder Tina und Jan

Inhaltsverzeichnis

Abbildungsverzeichnis	vi
Tabellenverzeichnis	xi
Abstract	xii
Nomenklatur	xiv
1 Einleitung	1
1.1 Motivation	1
1.2 Stand der Technik	2
1.3 Aufbau der Arbeit	5
2 Formulierung der Aufgabenstellung	6
2.1 Allgemeine Aufgabenstellung	6
2.2 Diagnoseverfahren auf der Basis multivariater Methoden	7
2.3 Konkretisierung der Aufgabenstellung	9
2.4 Zusammenfassung	11
3 Grundlagen datenbasierter Softwareinformationssysteme zur Anlagenüberwachung	12
3.1 Motivation	12
3.2 Datenvorverarbeitung	13
3.2.1 Problemformulierung	13
3.2.2 Behandlung von fehlenden Daten	14
3.2.3 Behandlung von Ausreißern	17
3.2.4 Zusammenfassung	19
3.3 Datenbasierte Methoden der Fehlererkennung und Überwachung	19
3.3.1 Fehlererkennung auf Basis der PCA	19
3.3.1.1 Das off-line-Training (Design)	20
3.3.1.2 Die on-line-Überwachung	25
3.3.2 Die dynamische PCA - DPCA	27
3.3.3 Rekursive PCA - RPCA	28
3.3.4 Kernel-PCA	30
3.3.5 Änderungsdetektion mit dem Generalized Likelihood Ratio (GLR)	32
3.3.6 Zusammenfassung	39
3.4 Anforderungen und Richtlinien an ein neues datenbasiertes Software- diagnose und -informationssystem	40

3.4.1	Motivation	40
3.4.2	Aufgabenstellung und Ziele	41
3.4.3	Anforderungen an das Design der Softwarearchitektur	44
3.4.3.1	Architektursichten	44
3.4.3.2	Modularität und Aufteilung in Schichten	45
3.4.3.3	Nutzung von Patterns und Frameworks	47
3.4.3.4	Konventionen der Softwareentwicklung	49
3.4.4	Zusammenfassung	50
3.5	Die Gurtförderanlagen als komplexes mechatronisches System	50
3.5.1	Grundlegender Aufbau	50
3.5.2	Der Prozess	53
3.5.3	Fehlerzustände	54
3.6	Zusammenfassung	55
4	Entwicklung datenbasierter Verfahren zur Anlagenüberwachung	56
4.1	Problemformulierung	56
4.2	Datenaufbereitung	57
4.2.1	Verminderung von Dynamiken im Erwartungswert durch lineare Kombination von Messgrößen	57
4.2.2	Anpassung der Varianz	71
4.3	Analyse der Zusammenhänge zwischen PCA und GLR	83
4.4	Adaptive GLR-Methode	85
4.4.1	Partielle adaptive GLR-Methode	85
4.4.2	Adaptive GLR-Methode mit Vergessensfaktor	86
4.4.3	Adaptive GLR-Methode mit einem gleitenden Fenster	87
4.5	Zusammenfassung	88
5	Entwurf des neuen Softwareinformationssystems	90
5.1	Motivation	90
5.2	Funktionelle Abläufe	93
5.3	Nutzung von Frameworks	93
5.4	Architekturentwurf	94
5.4.1	Client tier	95
5.4.2	Middle tier	96
5.4.3	Back end tier	100
5.5	Zusammenfassung	100
6	Anwendung der Verfahren an einer Gurtförderanlage	102
6.1	Beschreibung der Gurtförderanlage	102
6.2	Untersuchter Produktionsdatensatz	105
6.3	Fehlerzustände	107
6.3.1	Ausfall der Antriebstrommel 1 durch Wellenbruch	107
6.3.2	Mantelriss	108
6.3.3	Sensorfehler	108
6.3.4	Erhöhte Schwingungen an der Antriebstrommel 2	109

6.4	Zusammenfassung	110
7	Ergebnisse	111
7.1	Fehlerdetektion mithilfe der PCA	111
7.1.1	Ausfall der Antriebstrommel 1 durch Wellenbruch	112
7.1.2	Mantelriss	114
7.1.3	Sensorfehler	116
7.1.4	Erhöhte Schwingungen an der Antriebstrommel 2	118
7.2	Datenvorverarbeitung	119
7.2.1	Verminderung von Dynamiken im Erwartungswert durch lineare Kombination von Messgrößen	119
7.2.1.1	PCA mit einem Differenzsignal	121
7.2.1.2	PCA mit Differenzsignalen aus allen stark korrelierenden Signalen	124
7.2.2	Anpassung der Varianz	128
7.3	Fehlerdetektion mithilfe der Methode des GLR	130
7.3.1	Ausfall der Antriebstrommel 1 durch Wellenbruch	130
7.3.2	Mantelriss	132
7.3.3	Sensorfehler	133
7.3.4	Erhöhte Schwingungen an der Antriebstrommel 2	135
7.4	Diskussion der Ergebnisse	136
8	Zusammenfassung und Ausblick	138
Anhang A		143
A.1	Funktionelle Abläufe	143
Anhang B		146
B.1	Das Paket Data-Source	146
B.2	Das Paket Processing	147
B.3	Das Paket Action	149
B.4	Das Paket Job	151
B.5	Das Paket Context	153
	Literaturverzeichnis	156

Abbildungsverzeichnis

1.1	Grobe Einteilung der Fehlerdiagnosemethoden nach [14]	2
1.2	Schematische Beschreibung eine modellbasierten Fehlerdiagnosesystems nach [14]	3
1.3	Schematische Beschreibung einer Residuenauswertung und Entscheidungslogik basierend auf statistischen Tests nach [14]	4
2.1	Schema der Prozessüberwachung [7]	8
2.2	Übersicht über Methoden der Fehlererkennung	8
2.3	Grundstruktur eines mechatronischen Systems; Quelle: VDI-Richtlinie „Entwicklungsmethodik für mechatronische Systeme“, VDI 2206	10
3.1	Zusammenhang zwischen Boxplot und Dichtefunktion	18
3.2	Ablauf des PCA off-line Trainings	20
3.3	Hauptachsentransformation für ein zweidimensionales Beispiel [39] . .	23
3.4	Ablauf der on-line Überwachung	26
3.5	Konstruktion eines Neyman-Pearson Tests	35
3.6	Ablauf des GLR off-line Trainings	38
3.7	Ablauf der on-line Überwachung	38
3.8	Prinzipieller Aufbau des Teach-Moduls	42
3.9	Prinzipieller Aufbau des Runtime-Moduls	43
3.10	Ablauf - Überwachung und Wissensaufbau	43
3.11	Das 4+1 Sichten-Modell nach P. Kruchten [41]	44
3.12	Evolutionärer Prozess der Softwarearchitektur laut Bosch [4]	46
3.13	Aufbau einer Multi-Tier Architektur laut [9]	47
3.14	MVC Pattern	48
3.15	Schematische Darstellung einer Gurtförderanlage [35]	52
3.16	Modell einer Gurtförderanlage mit diskreten Massen aus [35]	53
3.17	Dynamischer Prozess mit ereignisbestimmten Arbeitspunktwechseln (Bsp. Gurtförderanlage)	54
3.18	Schematische Darstellung eines Prozesses unter Berücksichtigung von Fehlern	55
4.1	Zeitverlauf des Mittelwertes der Lagertemperatur Antriebstrommel 2 der Bandantriebsstation gegenüber dem Mittelwert des angelernten fehlerfreien Zustandes von Dezember bis Februar	58
4.2	Zeitverlauf ausgewählter Signale einer Bandantriebsstation	58
4.3	Prinzip der Differenzsignalbildung	60
4.4	Korrelationsmatrix für die Messsignale einer Bandantriebsstation . .	61

4.5	Daten mit aufgeprägtem multiplikativem Fehler	64
4.6	Daten mit aufgeprägtem additivem Fehler	64
4.7	Daten mit multiplikativem Fehler nach der Normierung von Mittelwert und Standardabweichung	64
4.8	Daten mit additivem Fehler nach der Normierung von Mittelwert und Standardabweichung	64
4.9	Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler ohne Dif- ferenzsignalbildung	65
4.10	Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler ohne Differenz- signalbildung	65
4.11	Daten mit multiplikativem Fehler und nicht normiertem Differenzsignal	65
4.12	Daten mit additivem Fehler und nicht normiertem Differenzsignal . .	65
4.13	Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler mit nicht normiertem Differenzsignal	66
4.14	Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler mit nicht nor- miertem Differenzsignal	66
4.15	Daten mit multiplikativem Fehler und normiertem Differenzsignal . .	66
4.16	Daten mit additivem Fehler und normiertem Differenzsignal	66
4.17	Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler mit nor- miertem Differenzsignal	67
4.18	Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler mit normiertem Differenzsignal	67
4.19	Daten mit aufgeprägtem multiplikativem Fehler auf x_1 und x_2 vor und nach der Normierung	68
4.20	Daten mit aufgeprägtem additivem Fehler auf x_1 und x_2 vor und nach der Normierung	68
4.21	Daten mit multiplikativem Fehler und normiertem Differenzsignal . .	68
4.22	Daten mit additivem Fehler und normiertem Differenzsignal	68
4.23	Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler mit nor- miertem Differenzsignal	69
4.24	Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler mit normiertem Differenzsignal	69
4.25	Daten mit aufgeprägtem multiplikativem Fehler auf x_1 und x_2 vor und nach der Normierung	70
4.26	Daten mit aufgeprägtem additivem Fehler auf x_1 und x_2 vor und nach der Normierung	70
4.27	Daten mit multiplikativem, nicht korrelierendem Fehler und normier- tem Differenzsignal	70
4.28	Daten mit additivem, nicht korrelierendem Fehler und normiertem Dif- ferenzsignal	70
4.29	Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler mit nor- miertem Differenzsignal	71
4.30	Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler mit normiertem Differenzsignal	71
4.31	Verlauf für Bandantriebsstation Leistung Motor 1	72

4.32	Diagramm Normalverteilung für Bandantriebsstation Leistung Motor 1	72
4.33	Histogramm für Bandantriebsstation Leistung Motor 1	73
4.34	Scatterplot für ausgewählte Daten der Bandantriebsstation vor dem Schneiden	73
4.35	Diagramm Bandantriebsstation Vibration Motor 1 vor dem Schneiden und dem zugehörigen Schnittindex	75
4.36	Diagramm Bandantriebsstation Vibration Motor 1 nach dem Schneiden	75
4.37	Diagramm Bandantriebsstation Leistung Motor 1 nach dem Schneiden	76
4.38	Histogramm für Bandantriebsstation Leistung Motor 1 nach dem Schnei- den	76
4.39	Diagramm Normalverteilung für Bandantriebsstation Leistung Motor 1 nach dem Schneiden	77
4.40	Scatterplot für ausgewählte Daten der Bandantriebsstation nach dem Schneiden	77
4.41	T^2 -Statistik für die Daten des Normalzustandes vor dem Schneiden .	78
4.42	SPE -Statistik für die Daten des Normalzustandes vor dem Schneiden	78
4.43	T^2 -Statistik für die Daten des Normalzustandes nach dem Schneiden	79
4.44	SPE -Statistik für die Daten des Normalzustandes nach dem Schneiden	79
4.45	Daten des fehlerfreien Zustandes mit Ausreißer bis zu 3σ	80
4.46	off-line-Training PCA zur Ermittlung der angepassten Kovarianzmatrix	81
4.47	off-line-Training GLR zur Ermittlung der angepassten Kovarianzmatrix	81
4.48	on-line-Überwachung (GLR und PCA) auf Basis der angepassten Ko- varianzmatrix	82
4.49	T^2 -Statistik für die Daten des Normalzustandes nach dem Schneiden	82
4.50	SPE -Statistik für die Daten des Normalzustandes nach dem Schneiden	83
5.1	Fehlerdetektionssystem auf der Basis multivariater Methoden	91
5.2	Multi tier Architektur	95
5.3	Aufteilung der Middle tier in Web tier und Business tier	95
5.4	Einsatzszenarien für die Middle tier	96
5.5	Aufteilung der Middle tier in Web tier und Business logic tier	97
6.1	Schematischer Aufbau der Bandantriebsstation	103
6.2	Gurtförderanlage Chile Kupferbergbau	105
6.3	Ausgewählte Variablen ohne Datenvorverarbeitung	107
6.4	Ausgewählte Variablen nach der Datenvorverarbeitung	107
7.1	Fehler Wellenbruch an der Antriebstrommel 1: Schwingungen an der Antriebstrommel 1 und 2	112
7.2	Fehler Wellenbruch an der Antriebstrommel 1: Strom und Leistung Motor 1	112
7.3	Fehler Wellenbruch an der Antriebstrommel 1: T^2 -Index nach der PCA	113
7.4	Fehler Wellenbruch an der Antriebstrommel 1: SPE -Index nach der PCA	113
7.5	Fehler Mantelriss an der Antriebstrommel 1: Vibrationen	114
7.6	Fehler Mantelriss an der Antriebstrommel 1: Strom und Leistung . .	115

7.7	Fehler Mantelriss an der Antriebstrommel 1: T^2 -Index nach PCA . . .	115
7.8	Fehler Mantelriss an der Antriebstrommel 1: SPE -Index nach PCA .	116
7.9	Fehler Sensorausfall: Vibrationsdaten	117
7.10	Fehler Sensorausfall: T^2 -Index nach der PCA	117
7.11	Fehler Sensorausfall: SPE -Index nach der PCA	117
7.12	Fehler Erhöhte Schwingungen an Antriebstrommel 2: Vibrationen an den Motoren der Antriebstrommeln 1 und 2	118
7.13	Fehler Erhöhte Schwingungen an Antriebstrommel 2: Strom Motor 1, Temperatur und Vibration Antriebstrommel 2	118
7.14	Fehler Erhöhte Schwingungen an Antriebstrommel 2: T^2 -Index nach der PCA	119
7.15	Fehler Erhöhte Schwingungen an Antriebstrommel 2: SPE -Index nach der PCA	119
7.16	Schwingungen Antriebstrommel links und rechts sowie Differenz zwi- schen beiden Signalen	120
7.17	Korrelationsmatrix gebildet aus den Trainingsdaten	121
7.18	Eigenwerte ohne Differenzbildung zwischen den Schwingungssignalen links und rechts, prozentual und absolut	122
7.19	Eigenwerte nach der Differenzbildung zwischen den Schwingungssigna- len links und rechts, prozentual und absolut	122
7.20	T^2 -Index ohne Differenzbildung	123
7.21	T^2 -Index nach der Differenzbildung zwischen den Schwingungssignalen links und rechts	123
7.22	SPE -Index ohne Differenzbildung	124
7.23	SPE -Index nach der Differenzbildung zwischen den Schwingungssi- gnalen links und rechts	124
7.24	Korrelationsmatrix gebildet aus den Trainingsdaten	125
7.25	Eigenwerte ohne Differenzbildung	125
7.26	Eigenwerte nach der Differenzbildung zwischen mehreren Signalen . .	126
7.27	T^2 -Index ohne Differenzbildung	126
7.28	T^2 -Index nach der Differenzbildung zwischen mehreren Signalen . . .	127
7.29	SPE -Index ohne Differenzbildung	127
7.30	SPE -Index nach der Differenzbildung	127
7.31	Fehler: Wellenbruch an Antriebstrommel 1; Eigenwerte	129
7.32	Fehler: Wellenbruch an Antriebstrommel 1; Eigenwerte nach Neubil- dung der Kovarianzmatrix	129
7.33	Fehler: Wellenbruch an Antriebstrommel 1; T^2 -Index nach Neubildung der Kovarianzmatrix	129
7.34	Fehler: Wellenbruch an Antriebstrommel 1; SPE -Index nach Neubil- dung der Kovarianzmatrix	130
7.35	Fehler: Wellenbruch an Antriebstrommel 1; T^2 -Index nach GLR . . .	131
7.36	Fehler Wellenbruch an Antriebstrommel 1: T^2 -Index nach GLR; Schwell- wert aus maximalem T^2 -Index der Trainingsdaten	132

7.37 Fehler: Wellenbruch an Antriebstrommel 1; T^2 -Index nach GLR; Schwellwert aus maximalem T^2 -Index der Trainingsdaten und angepasster Kovarianzmatrix	132
7.38 Fehler Mantelriss Antriebstrommel 1: T^2 -Index nach GLR	133
7.39 Fehler Mantelriss Antriebstrommel 1: T^2 -Index nach GLR; Schwellwert aus maximalem T^2 -Index der Trainingsdaten und angepasster Kovarianzmatrix	133
7.40 Fehler Sensorausfall: 1; T^2 -Index nach GLR	134
7.41 Fehler Sensorausfall: T^2 -Index nach GLR; Schwellwert aus maximalem T^2 -Index der Trainingsdaten und angepasster Kovarianzmatrix	134
7.42 Fehler Erhöhte Schwingungen an Antriebstrommel 2: T^2 -Index nach GLR	135
7.43 Fehler Erhöhte Schwingungen an Antriebstrommel 2: T^2 -Index nach GLR; Schwellwert aus maximalem T^2 -Index der Trainingsdaten und angepasster Kovarianzmatrix	135
A.1 Funktioneller Ablauf der Software aus [21]	143
A.2 Funktioneller Ablauf zur Charakterisierung des Prozesses aus [21]	144
A.3 Funktioneller Ablauf Datenvorverarbeitung aus [21]	145
B.1 Klassendesign Data-Source aus [21]	146
B.2 Klassendesign für das Paket Processing aus [21]	149
B.3 Klassendesign Ereignisbehandlung im Paket Action [21]	150
B.4 Klassendesign Job aus [21]	151
B.5 Klassendesign Math-Service aus [21]	155

Tabellenverzeichnis

3.1	Kombinationen - wahrer Zustand, Entscheidung und Folgen	34
6.1	Liste verwendeter Variablen der Gurtbandförderanlage CV01 nach Angaben des Anlagenherstellers	106

Abstract

Die vorliegende Arbeit beinhaltet Aufgabenstellungen aus dem Gebiet der multivariaten Fehlerdetektion an mechatronischen Systemen. Die entwickelten Verfahren erlauben, über das Anlernen des fehlerfreien Zustandes und Generieren eines Modells im statistischen Sinne die Fehlerdetektion an einem komplexen Prozess durchzuführen. Der Nachweis der Anwendbarkeit der vorgeschlagenen Methoden erfolgt an einem komplexen mechatronischen System. Als Benchmark dient ein Massentransportprozess aus dem Kupferbergbau. Zur Vorbereitung der Integration der entwickelten Verfahren in ein neues Softwaremodul des Asset-Management-Systems zedas®asset wird das Design der neuen Softwarearchitektur vorgestellt. Mit der Anwendung der beiden Schritte off-line-Training und on-line-Überwachung an dem mechatronischen System Gurtförderanlage, mit vier im praktischen Betrieb der Anlage aufgetretenen Fehlerzuständen, erfolgt die Bestätigung der Funktionalität der entwickelten Algorithmen.

The following work contains tasks in the field of multivariate fault detection in mechatronic systems. By learning of the fault-free state and generating a model in a statistical sense, the developed methods allow the performance of fault detection in a complex process. The proof of the applicability of the proposed methods is demonstrated by a complex mechatronic system. The mass transport process from copper mining serves as a benchmark. In preparation for the integration of the developed methods into a new software module of the Asset-Management-Systems zedas®asset the design of the new software architecture is introduced. By applying both off-line training and on-line monitoring of the mechatronic system belt conveyor, with four of the plants practical operation error states occurring, the functionality of the developed algorithms is confirmed.

Nomenklatur

Die wichtigsten Formelzeichen in dieser Arbeit werden wie folgt definiert:

Mathematische Formelzeichen

\mathbf{R}	: Menge der reellen, komplexen Zahlen
\mathbf{R}^n	: Menge der n -dimensionalen reellen, komplexen Vektoren
$\mathbf{R}^{n \times m}$: Menge der $n \times m$ -dimensionalen reellen, komplexen Matrizen
$\mathcal{N}, \mathcal{N}_m$: Normalverteilung, m -dimensionale Multinormalverteilung
I	: Einheitsmatrix
$x, x(t), X$: Vektor, Vektor der Zeitfunktion, nicht normierte Datenmatrix
o, O	: Nullvektor bzw. -matrix
X^\top, X^{-1}	: transponierte, inverse Matrix zu X
Z	: normierte Ausgangsdatenmatrix
$Z(h)$: Hankel-Matrix von Z
h	: Anzahl der lags
N	: Anzahl der Elemente (Samples) von Z
m	: Anzahl der Spalten von Z
$\lambda(z)$: Eigenwert der Matrix Z
σ	: Standardabweichung
σ_k	: Standardabweichung zum Zeitpunkt k
σ^2	: Varianz
1_N	: Vektor mit Wert 1 in jedem der N Elemente
$E(Z_0)$: Erwartungswert von Z_0
μ_k	: Mittelwert zum Zeitpunkt k
Σ	: Kovarianzmatrix
P	: Eigenvektoren
Λ	: Eigenwerte
Λ_0	: Eigenwerte nach dem off-line Training
l	: Anzahl der Hauptkomponenten
l_0	: Anzahl der Hauptkomponenten nach dem off-line Training
P_{pc}	: Eigenvektoren mit dem größten Varianzanteil (Hauptkomponenten)
P_{res}	: Eigenvektoren mit dem geringen Varianzanteil (residual space)
P_0	: Hauptkomponenten nach dem off-line Training
μ	: Mittelwert
J_{th, T^2}	: T^2 Schwellwert

$J_{th,SPE}$:	SPE Schwellwert
SPE	:	SPE Index (SPE - squared prediction error)
T^2	:	T^2 Index
F_α	:	F-Verteilung für das vorgegebene Signifikanzniveau α
χ^2	:	Chi-Quadrat-Verteilung
$T^2(m,N)$:	Hotellings T^2 -Verteilung mit den Freiheitsgraden m und N
$k(z_i, z_j)$:	Kernelfunktion
C	:	Mahalanobis-Distanz
$L(X; \Theta)$:	Likelihood-Funktion von Θ
$l(X; \Theta)$:	logarithmierte Likelihood-Funktion (Log-Likelihood-Funktion) von Θ
H_0, H_1	:	Nullhypothese, Alternativhypothese
$lq(x)$:	Log-Likelihood-Quotient für x
$g(x)$:	Generalized-Likelihood-Quotient für x
$p(x)$:	Wahrscheinlichkeit von x
S	:	normierte Varianz-Kovarianz-Matrix
C	:	Mahalanobis-Distanz
MDR	:	Missdetektionsrate
FAR	:	Fehlalarmrate
ε	:	zufälliger unkorrelierter Signalanteil
\in	:	Element von
$ \cdot $:	Betrag
$\ \cdot\ $:	euklidische Vektornorm
\Longleftrightarrow	:	genau dann, wenn
\forall	:	für alle

Symbolverzeichnis für Gurtförderanlagen

Symbol	Einheit	Bezeichnung
M_i	: $[Nm]$	Motormoment i
$m_{A,U,i}$: $[kg]$	auf den Fördergurt reduzierte Masse
$M(t)$: $[Nm]$	Motormoment
$Q(t)$: $[kg/s]$	Massenstrom
$I(t)$: $[A]$	Motorstrom
$n(t)$: $[min^{-1}]$	Drehzahl
$T_{sp}(t)$: $[N]$	Gurtkraft an der Spannstelle
$V_a(t)$: $[m/s^2]$	Trendwert aus der Vibrationsanalyse

Verwendete Abkürzungen

DTO	: Data Transfer Object
FDI	: Fault Detection and Isolation
FDA	: Fisher Discriminant Analysis
GUI	: Graphical User Interface
Java-EE	: Spezifikation einer Softwarearchitektur für die transaktionsbasierte Ausführung in Java dokumentiert in [11]
JBoss	: Enterprise Middleware nach Java-EE-Standard der Firma Red Hat
JSF	: Java Server Faces dokumentiert in [12]
JSP	: Java Server Pages - Web-Programmiersprache entwickelt von Sun Microsystems
MVC	: Model-View-Controller
EJB	: Enterprise Java Bean
REST	: Representational State Transfer
PCA	: Principal Component Analysis
OLE	: Object Linking and Embedding
OPC	: OLE for Process Control
OMG	: Object Management Group
OCSVM	: One Class Support Vector Machine
SVM	: Support Vector Machine
zedas®asset	: Anlagen Management System der Firma PC-Soft GmbH

1 Einleitung

1.1 Motivation

Die steigende Komplexität technischer Prozesse und Anlagen, verbunden mit einer Erhöhung des Integrationsgrades, führt oft zu einer gesteigerten Sensitivität dieser Systeme. Aus diesem Grund stellt die Gewährleistung einer hohen Verfügbarkeit neben der Verbesserung der Anlagen- oder Systemperformanz ein wesentliches Entwicklungsziel dar. Fehler, Ausfälle oder erhöhte Risiken müssen somit sicher und rechtzeitig diagnostiziert werden, um der Prozessführung eine angemessene Reaktion zu ermöglichen oder erforderliche Wartungsarbeiten anzuzeigen. Die Qualität komplexer technischer Prozesse wird deshalb zunehmend auch an den Fähigkeiten der Diagnose- und Überwachungsfunktionen gemessen. Für die Industrie sind in diesem Zusammenhang insbesondere Ansätze interessant, die ohne oder nur mit geringfügigem gerätetechnischen Mehraufwand gewonnen werden können. Somit ist es möglich, zusätzliche Informationen ohne ergänzende Sensoren und den damit verbundenen Kosten für die Beschaffung und den Einbau zu erhalten. In der Anlagenindustrie genießen datengestützte, statistische Verfahren zur Überwachung und Fehlerdiagnose eine sehr hohe Akzeptanz. Der Vorteil dieser Verfahren zur Fehlerdiagnose basiert auf einem Modell, im Sinne von statistischen Eigenschaften, welches direkt aus den Prozessmessdaten und ohne spezifische regelungstechnische Kenntnisse gewonnen werden kann. Ein Nachteil dieser Realisierung ist, dass die Prozessdynamiken oder die Nichtlinearitäten im Prozessverhalten bisher nicht genügend berücksichtigt wurden. Des Weiteren fehlen in der Industrie bisher Softwareprodukte, welche diese Verfahren zur Prozessüberwachung und Fehlerdiagnose nutzen. Motiviert durch die Ergebnisse vorangegangener Forschungsprojekte wurden im Rahmen dieser Arbeit datengestützte Verfahren zur Fehlererkennung untersucht. Den Schwerpunkt der vorliegenden Arbeit bilden Erkenntnisse der theoretischen Untersuchung sowie die Implementierung, Simulation und Realisierung der neuen Verfahren unter industriellen Bedingungen am Beispiel eines komplexen Massentransportsystems. Der nachfolgende Abschnitt soll einen kurzen Überblick über den aktuellen Stand der Forschung und Entwicklung auf dem Gebiet der datengestützten Prozessüberwachung und Fehlerdiagnose geben.

1.2 Stand der Technik

Ein System zur Fehlerdiagnose hat im Wesentlichen folgende Aufgaben zu erfüllen:

- Fehlerdetektion - Erkennen des Andeutens und Auftretens von Fehlerzuständen an den Komponenten der überwachten Anlage, welche zu kritischen Betriebszuständen während des Prozesses führen können
- Fehlerisolation - Lokalisierung und Klassifizierung unterschiedlicher Fehler
- Fehleranalyse - Bestimmung der Fehlergröße und der Ursache für den Fehler.

Ein nach diesem Prinzip arbeitendes technisches System, nennt man FDI (fault detection and isolation) - System.

Ein FDI-System erledigt seine Aufgaben in zwei Teilschritten:

- Fehlererkennung:
Erzeugen von Vergleichsgrößen aus den Messdaten; Nachweis eines Fehlers und ermitteln von Symptomen mit Hilfe von nominalen Größen
- Fehlerdiagnose:
Isolieren und identifizieren des Fehlers; Bestimmung von Typ, Ausmaß, Ort und Zeit des Auftretens des wahrscheinlichsten Fehlers aus den bereitgestellten Symptomen und der Kenntnis der Beziehung zwischen Symptomen und den Fehlern.

In [14] werden die Methoden der Fehlerdiagnose, wie in der Abbildung 1.1 dargestellt, eingeteilt.

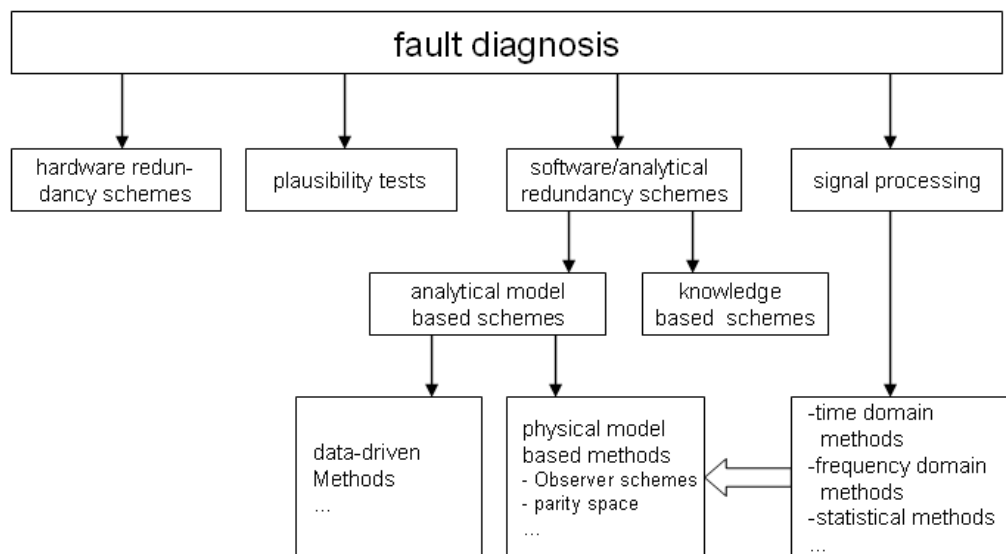


Abb. 1.1: Grobe Einteilung der Fehlerdiagnosemethoden nach [14]

Venkatasubramanian differenzierte in [71], [70], [72] nach der Art des Wissens, ob es sich um a priori oder historisches Wissen handelt. Modellbasierte Methoden erfordern a priori Wissen um, den modellierten Fehler abzuschätzen, während die datenbasierten Methoden das Wissen aus quantitativen historischen Prozessdaten gewinnen.

In [14] und [35] haben die Autoren die modellbasierten Methoden zur Fehlerdiagnose hinreichend untersucht und beschrieben. Die nachfolgende Abbildung zeigt den prinzipiellen Aufbau eines modellbasierten Fehlerdiagnosesystems.

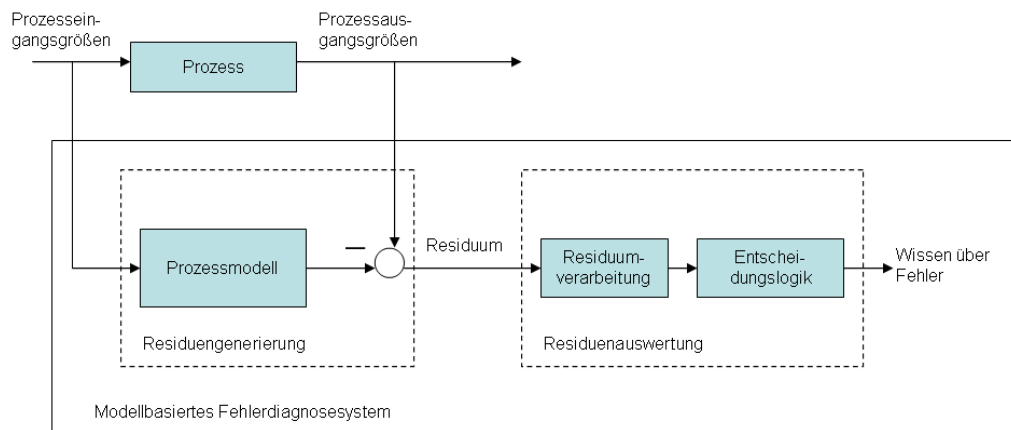


Abb. 1.2: Schematische Beschreibung eines modellbasierten Fehlerdiagnosesystems nach [14]

Erfahrungen der letzten Jahre haben gezeigt, dass der Einsatz der verfügbaren modellbasierten Methoden unter Praxisbedingungen erschwert wird, weil die meisten Anwender wenige Kenntnisse über die entwickelten Methoden besitzen, es wenige rechnergestützte Werkzeuge gibt, der Aufwand zur Erstellung der Modelle verhältnismäßig groß ist und eine Wiederverwendbarkeit meist nicht gegeben ist.

In den vergangenen Jahren wurde insbesondere der Entwicklung von datengestützten Verfahren zur Prozessüberwachung und Fehlerdiagnose große Aufmerksamkeit zuteil. Ihre wesentliche Idee ist in der Abbildung 1.2 dargestellt. Der prinzipielle Aufbau ähnelt dem eines modellbasierten Fehlerdiagnosesystems und beinhaltet, dass zunächst aus den Prozesseingangs- und -ausgangsgrößen des fehlerfreien Zustandes ein Modell im statistischen Sinne sowie die zugehörigen Schwellwerte generiert werden. Während der on-line-Überwachung berechnet der Residuengenerator aus den aktuellen Prozessmessdaten und den statistischen Modellparametern die Residuen. Die danach stattfindende Residuenauswertung beinhaltet den Vergleich der Residuen mit den Schwellenwerten (siehe [61], [54] und [14]).

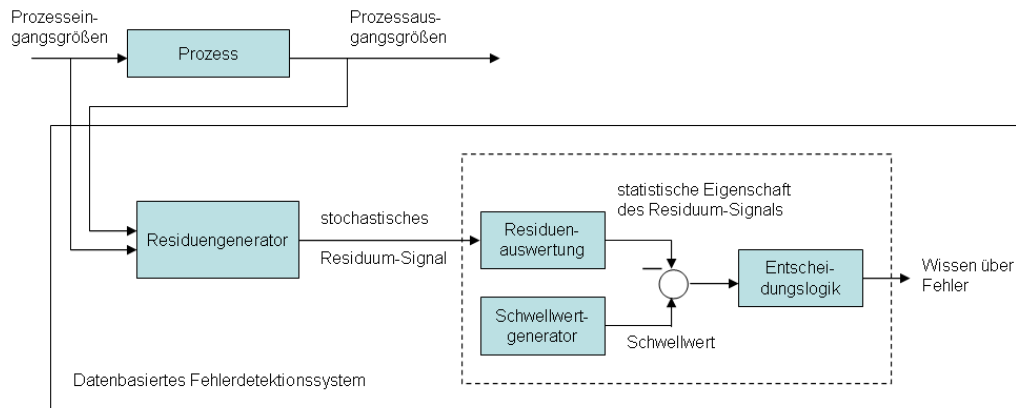


Abb. 1.3: Schematische Beschreibung einer Residuenauswertung und Entscheidungslogik basierend auf statistischen Tests nach [14]

Bei datenbasierten Methoden werden mehrdimensionale Daten, die Aufschluss über den Zustand des Prozesses geben, direkt verarbeitet. Ein großer Vorteil dieser Methoden gegenüber den modellbasierten ist, dass sie direkt auf komplexe Prozesse angewendet werden können. Bei modellbasierten Methoden bedingt das die Generierung eines analytischen Modells, was für solche Prozesse mit sehr großem Aufwand verbunden, wenn nicht gar unmöglich ist. Ein Nachteil der datenbasierten Methoden stellt die direkte Abhängigkeit der Effizienz von der Quantität und Qualität der Prozessdaten dar, welche für die Analyse verwendet wurden. Die Entwicklungen insbesondere auf dem Gebiet der Hauptkomponentenanalyse (engl. Principal Component Analysis - PCA) wurden durch die Arbeiten von Jolliffe [38] in den achtziger Jahren sowie durch Russel, Chiang und Braatz [58] geprägt. Die verwendeten T^2 und SPE Statistiken wurden unter anderem in [58] beschrieben, wobei die Grundlagen dafür Hotelling [29] und Box [6] lieferten. In der Vergangenheit gab es eine Reihe von Abwandlungen zur Fehlerdetektion auf Basis der PCA, z. B. die rekursive PCA, zur Anwendung bei langsamen Änderungen von Prozessparametern, beschrieben in [45], die dynamische PCA vorgestellt in [59], welche bei Auftreten von seriellen Korrelationen, wie sie bei dynamischen Systemen auftreten, angewendet wird oder die Kernel-PCA, vorgestellt und verallgemeinert in [60] für die Fehlerdetektion bei nicht-linearen Prozessen. Basseville und Nikiforov stellten in [2] datenbasierte Methoden zur Fehlerdetektion vor, die auf dem „Generalized Likelihood Ratio“ beruhen.

Der Stand der Technik ermöglicht die Anwendung einer Reihe von bekannten datenbasierten multivariaten Methoden zur Fehlerdetektion, welche größtenteils auf der PCA fußen. Die Grundlage für die Anwendung datenbasierter Methoden ist in der Praxis zumeist gegeben, nämlich Prozessdatenerfassungssysteme, welche die Daten von den Prozesssteuerungs- und Leitsystemen übernehmen, archivieren und zur Auswertung übergeordneten Systemen bereitstellen. Für den Anwender stellt sich aber die Frage: Wie kann das Bedienpersonal die datenbasierten Methoden einfach anwenden, ohne vertiefendes mathematisches und statistisches Grundwissen und ohne allzu tiefe Kenntnisse über den eigentlichen Prozess zu besitzen? Um dieser Anforderung

gerecht zu werden, sollen die datenbasierten multivariaten Methoden in einem leistungsfähigen Softwareinformationssystem zur Anlagenüberwachung eingesetzt werden, welches eine effektive automatisierte, aber einfach anzuwendende Überwachung und Fehlererkennung ermöglicht.

1.3 Aufbau der Arbeit

Nach der Konkretisierung der Aufgabenstellung dieser Arbeit in Kapitel 2 werden in Kapitel 3 wesentliche Grundlagen verschiedener datenbasierter multivariater Methoden zur Fehlerdetektion erläutert. Jeder dargelegten Methode folgt eine Beschreibung des Konzeptes zur Aufbereitung der Daten, zur automatisierten Bildung von Grenzwerten und zur Berechnung der Kennwerte. Aufgrund der Bedeutung der multivariaten Normalverteilung für den erfolgreichen Einsatz der Methoden enthält das Kapitel ebenfalls eine Darstellung bekannter Verfahren zur Datenvorverarbeitung. Das Ziel des gesamten Entwicklungsprozesses ist die Entwicklung eines Softwareinformationssystems zur Anlagenüberwachung. Im Rahmen dieser Arbeit sind die Grundlagen für das Design zu legen. Deshalb folgt ein Abschnitt mit den wesentlichen Anforderungen und Richtlinien, die im Designprozess eines Softwaresystems mit einer modernen Architektur zu beachten sind. Die entwickelten und angepassten Verfahren sollen an komplexen mechatronischen Systemen zur Fehlererkennung zum Einsatz kommen. Ein Schüttgut-Materialtransportprozess dient als Benchmark-Prozess und wird im letzten Abschnitt des Kapitel 3 erläutert.

Ausgehend von den in Kapitel 3 untersuchten multivariaten Methoden, widmet sich der Kern des 4. Kapitels der Darstellung neuer und erweiterter Fehlerdetektionsverfahren, welche die in Kapitel 2 beschriebenen Anforderungen erfüllen. Den Erläuterungen der neuen Verfahren folgen in Kapitel 5 die Beschreibungen der funktionellen Abläufe und des Architekturentwurfes des neuen Softwareinformationssystems. Das Kapitel 6 enthält eine konkrete Darstellung der als Benchmark-System eingesetzten Gurtförderanlage sowie der zur Überwachung zur Verfügung stehenden Prozessdaten und der darin enthaltenen Fehlerzustände.

In Kapitel 7 wird das vorgeschlagene Konzept, die entwickelten Algorithmen zur Überwachung des in Kapitel 6 beschriebenen industriellen Massentransportprozesses eingesetzt und die daraus resultierenden Untersuchungsergebnisse erläutert und illustriert. Die Eignung der entwickelten Methoden zur industriellen Anwendung wird nachgewiesen.

Zuletzt erfolgt eine Zusammenfassung der erreichten Ergebnisse sowie ein Ausblick, in welche Richtung die Forschung und Entwicklung fortgeführt werden könnte.

2 Formulierung der Aufgabenstellung

2.1 Allgemeine Aufgabenstellung

In der modernen Produktionstechnik werden immer komplexere hoch automatisierte Maschinen und Anlagen eingesetzt, um den steigenden Qualitätsanforderungen gerecht zu werden. Eine Folge dieser Entwicklung ist, dass die Analyse und Bewertung von Fehlentwicklungen entsprechend komplex und schwierig ist. Zwar stehen durch den zunehmenden Automatisierungsgrad der Anlagen immer größere Mengen an Sensor- und Anlagendaten zur Verfügung, für deren produktionsspezifische Aufbereitung und Interpretation gibt es jedoch kaum echtzeitfähige Werkzeuge und Modelle. Infolge dessen stehen dem Ingenieur zur Auswertung oftmals riesige unbearbeitete Datenarchive zur Verfügung. Die Messdatenarchivierung wird einerseits vom Gesetzgeber gefordert, um eine bestimmungsgemäße und sichere Fahrweise der Anlagen entsprechend der Errichtungsgenehmigung überprüfen zu können. Andererseits lassen sich die Daten somit über einen längeren, teilweise auch lebenslangen Zyklus einer Anlage gewinnbringend nutzen, etwa zur Prozessoptimierung, Fehlerdiagnose oder zur vorbeugenden Instandhaltung. In der Praxis werden diese Möglichkeiten bisher unzureichend ausgeschöpft. Dafür sind folgende Ursachen bekannt:

- Die Datenmengen sind zu groß, um sie im Nachhinein auszuwerten.
- Die herkömmlichen Methoden zur Online-Datenauswertung sind zu betreuungsintensiv.
- Durch die immer komplexer werdenden Anforderungen an das Service-Personal fehlt diesem die Zeit für eine umfangreiche Datenauswertung.

Ziel dieser Arbeit ist es, multivariate statistische Verfahren, die bereits erfolgreich in der Ökonomie zur Auswertung großer Datenmengen zum Einsatz kommen, zur Überwachung und Fehlerdiagnose von modernen Produktionsanlagen zu nutzen, wobei der gesamte Prozess zur Überwachung und Fehlererkennung bezüglich seines Betreuungsaufwandes minimal sein soll. Dies gilt für das

- Anpassen der Überwachungsalgorithmen
- Finden von Grenzwerten und
- Auswerten der Daten.

Die Aufgabenstellung dieser Arbeit resultiert aus den Anforderungen der Industrie. Zu diesen gehören unter anderem:

- Erhöhung der Verfügbarkeit von Maschinen und Anlagen
- Senkung der Instandhaltungsaufwendungen/ -kosten
- Verbesserung der Produktqualität
- Erhöhung der Sicherheit von Anlagen und Prozessen.

Werden Störungen, Fehler und Verschleiß unter den teilweise extremen Bedingungen eines realen Produktionsprozesses nicht rechtzeitig erkannt, führt das zwangsläufig zu Produktionsausfall, zu Schädigung von Material und damit verbundenen hohen Kosten. Dem versucht man in der Industrie entgegen zu wirken, indem Anlagen und Komponenten nach ihrer Bedeutung für die Verfügbarkeit und Sicherheit eingeschätzt sowie die daraus folgende Instandhaltungsstrategie zugeteilt wird. Für Verfügbarkeits- und sicherheitsrelevante Teile wird oftmals eine präventive oder auch zustands- bzw. belastungsabhängige Instandhaltungsstrategie ausgewählt. Zur Umsetzung dieser Strategien benötigt man in den meisten Fällen spezielle Diagnosesysteme (Schwingungsdiagnose, Thermografie, Ultraschallmessungen, Öl-Analyse-Systeme, ..., die in der Praxis oftmals unter dem Begriff „Condition Monitoring“ Systeme zusammengefasst werden) sowie gut ausgebildetes und meist speziell geschultes Personal. In beiden Fällen ist das ebenfalls mit vergleichsweise hohen Kosten verbunden. Daher verlangt die Industrie, nach

- einfach zu handhabenden, aussagekräftigen Diagnosesystemen
- Nutzung der vorhandenen Sensorik
- geringen Anforderungen an die Prozesskenntnis und an den Betreuungsaufwand für solche Systeme und
- sichere Detektion von Fehlern.

Die in der vorliegenden Arbeit dargestellten Methoden sollen diese Anforderungen erfüllen. Im Rahmen weiterer Forschungs- und Entwicklungsvorhaben sollen diese zusammen mit weiteren modell- und datenbasierten Anlagen- und Prozessüberwachungsmethoden in ein Softwaremodul integriert werden, das eine ganzheitliche Systemlösung für die Integration der auf verschiedenen Ebenen der Prozess- und Anlagenautomation verteilten Anlagenüberwachungs- und Diagnosefunktionalitäten ermöglicht.

2.2 Diagnoseverfahren auf der Basis multivariater Methoden

Die nachfolgende Grafik nach [7] verdeutlicht die grundlegenden Aufgaben, die ein Prozessüberwachungs- und Diagnosesystem zu erfüllen hat.

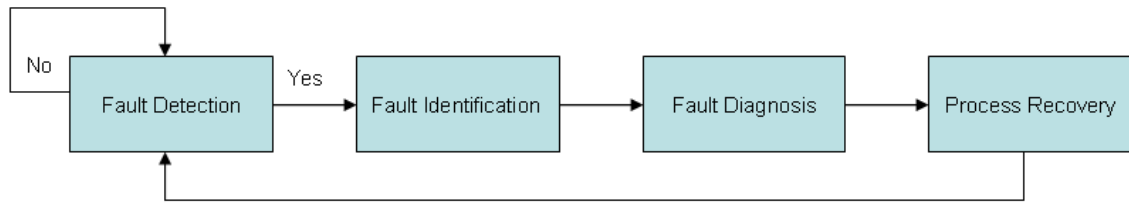


Abb. 2.1: Schema der Prozessüberwachung [7]

Unter Fehlererkennung versteht man die Feststellung der Präsenz eines Fehlers im System sowie der Zeit der Erkennung. Die Identifikation beinhaltet die Ermittlung der Fehlergröße, des Fehlerortes und des Zeitvarianz-Verhaltens des Fehlers. Mit der Fehlerdiagnose kann die Art des Fehlers, eventuell auch die Ursache benannt werden. Der Fehlerdiagnose folgt das Wiederherstellen des normalen Prozessverhaltens. Die Qualität rechnergestützter Fehlerdiagnosesysteme wird durch ihre Fehlalarmrate sowie durch die Fähigkeit, Fehler im Prozess frühzeitig zu erkennen, bestimmt.

In der aus [30] entnommenen Übersicht Abbildung 2.2 ist ersichtlich, dass es viele verschiedene Ansätze zur Fehlererkennung und Symptomgewinnung gibt. Alle Methoden eint jedoch dieselbe Grundidee: Aus Wissen und Beobachtungen über den Prozess sollen für den Zustand charakteristische Größen bestimmt und mit den jeweils erwarteten Werten verglichen werden.

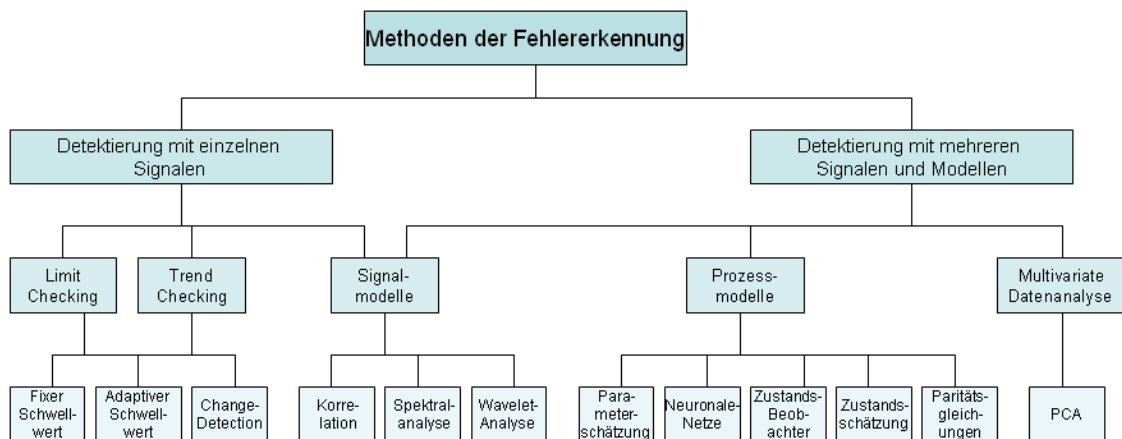


Abb. 2.2: Übersicht über Methoden der Fehlererkennung

Mittlerweile existieren eine Reihe von Methoden zur Fehlererkennung. In der Übersicht - Abbildung 2.2 sind einige der bekanntesten aufgeführt. Die Abbildung soll darüber hinaus verdeutlichen, wo sich die multivariaten Methoden einordnen. Des Weiteren arbeiten diese Methoden zwar nicht mit analytischen Modellen, die extrahierten Merkmale spiegeln jedoch wesentliche statistische Zusammenhänge wider, sodass man auch von statistischen Modellen sprechen kann.

Das Modell, ob nun analytisch- oder statistisch ermittelt, ist neben dem Residuengenerator und der Auswerteeinheit ein wesentlicher Bestandteil eines, wie in Abbildung 1.2 dargestellten Fehlerdiagnosesystems.

Ein analytisch gewonnenes Prozessmodell beruht auf physikalischen und mathematischen Gesetzmäßigkeiten (quantitatives Prozessmodell), während das statistisch gewonnene Wissen über den Prozess nur qualitativ angegeben werden kann (qualitatives Prozessmodell). Die aus dem Prozessmodell gewonnenen redundanten Variablen werden mit den realen Prozessführungsgrößen und Prozessmessgrößen verglichen. Die daraus gewonnene Differenz ist das Residuum. Die Gestaltung der Residuenauswerteeinheit hängt von der jeweiligen Aufgabe im Rahmen der Prozessüberwachung (Fehlerdetektion, Fehleridentifikation, siehe Abbildung 2.1) ab.

Nicht alle im realen Industrieprozess auftretenden Prozess- und Umgebungsbedingungen lassen sich messen. Die Anzahl der eingesetzten Sensoren ist aus technischen und wirtschaftlichen Gründen begrenzt. Somit kann das Modell den realen Prozess nur unvollständig nachbilden bzw. nur begrenzt auf Veränderungen reagieren. Als Folge muss die Auswerteeinheit ein gewisses Maß an Robustheit gegenüber Modellungenauigkeiten und Störgrößen aufweisen.

2.3 Konkretisierung der Aufgabenstellung

Schlussfolgernd aus den industriellen Anforderungen soll mit dieser Arbeit nachfolgende Fragestellung beantwortet werden:

Können multivariate statistische Verfahren zur Fehlerdetektion an industriellen Prozessen eingesetzt, somit die Verfügbarkeit, Zuverlässigkeit und Sicherheit von Massentransportprozessen verbessert sowie die Anforderungen an Einfachheit und Robustheit erfüllt werden?

Ein daraus resultierender Aufgabenblock ist folgender:

- Entwicklung/Anpassung multivariater Verfahren zur Fehlerdetektion an einem mechatronischen System unter Beachtung der industriellen Anforderungen
- Extraktion von statistischen Merkmalen und Bildung von Schwellwerten
- Anwendung und Validierung des vorgeschlagenen Konzeptes zur Überwachung eines industriellen Massentransportprozesses.

Ein weiteres Ziel ist die Klärung der Frage, ob die an einer realen Industrieanlage zur Verfügung stehenden Daten den Anforderungen zur Anwendung datenbasierter Verfahren entsprechen bzw. wie kann man die Daten in jenen Zustand versetzen, der optimale Diagnoseergebnisse erwarten lässt?

Daraus folgt die Aufgabe, bekannte Methoden der Datenaufbereitung zu untersuchen und bei Sinnfälligkeit diese an Daten eines mechatronischen Systems anzuwenden bzw. ergänzende Methoden zu entwickeln. Nachfolgend wird das mechatronische

System allgemein und später in einer speziellen, zur Anwendung der entwickelten Methoden im Rahmen dieser Arbeit eingesetzten Ausprägung, kurz erläutert.

Als mechatronisches System versteht man ein Gesamtsystem, welches durch die Zusammenführung von Teilsystemen bzw. Komponenten einzelner Teildisziplinen gebildet wird. Laut Isermann ist die Mechatronik ein „interdisziplinäres Gebiet, bei dem folgende Disziplinen zusammenwirken“[31]:

- Mechanische Systeme
- Elektronische Systeme
- Informationstechnik.

Laut der VDI-Richtlinie 2206 besteht ein mechatronisches System aus vier- meist einen Regelkreis bildende Funktionsgruppen:

- Grundsystem (mechanische, elektromechanische, hydraulische, pneumatische Komponenten oder aus einer Kombination dieser)
- Sensoren (physisch vorhandene Messwertaufnehmer oder Softwaresensoren)
- Informationsverarbeitung (dient der Beeinflussung von Zustandsgrößen, erfolgt digital und oder analog)
- Aktoren (setzen die Einwirkungen am Grundsystem).

Des Weiteren ist die Energieversorgung der mechanischen Haupt- und regelungstechnischen Hilfssysteme für ein mechatronisches System von Bedeutung. Die nachfolgende Abbildung 2.3 stellt den grundlegenden Aufbau grafisch dar.

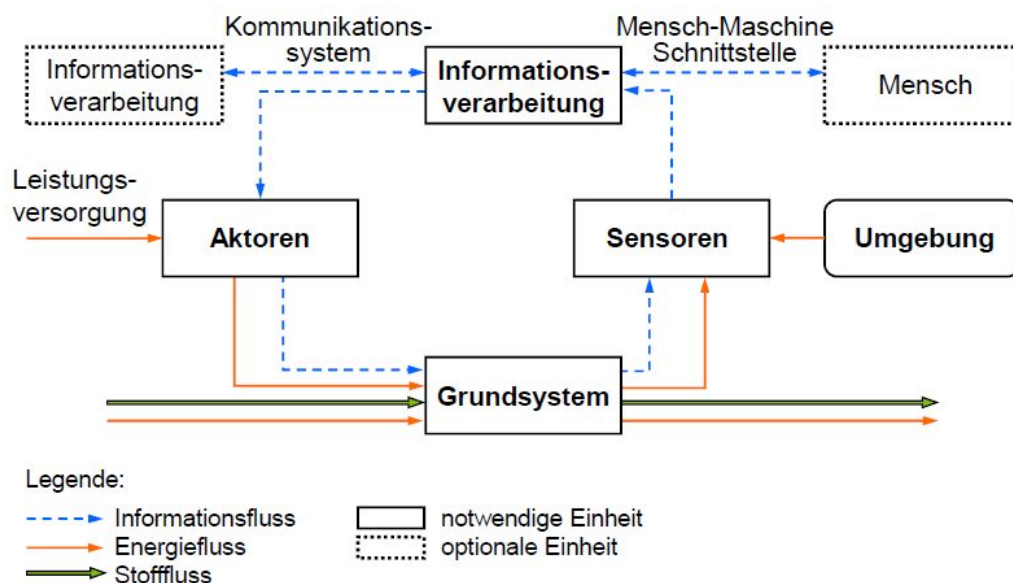


Abb. 2.3: Grundstruktur eines mechatronischen Systems; Quelle: VDI-Richtlinie „Entwicklungsmethodik für mechatronische Systeme“, VDI 2206

2.4 Zusammenfassung

Im Rahmen dieser Arbeit sind bekannte multivariate Methoden zu analysieren bzw. so weiterzuentwickeln, dass sie als Bestandteile von Diagnoseverfahren für die datenbasierte Fehlererkennung an mechatronischen Systemen geeignet sind. Anschließend erfolgt die Anwendung dieser Verfahren an einem ausgewählten mechatronischen System, einer Gurtförderanlage. Die Unterscheidung von stationärem und dynamischem Verhalten an konkreten mechatronischen Anlagen hat Bedeutung für die Fehlerdetektion und die Wahl des geeigneten datenbasierten Verfahrens zur Fehlerdetektion. Fehlerzustände und Störungen sollen ohne vorherige analytische Beschreibung sowohl der Anlage als auch der typischen und außergewöhnlichen Fehler erkannt werden. Es gibt eine Reihe von äußeren Einflüssen, welche das Verhalten einer Anlage beeinflussen, aber nicht unbedingt zu kritischen Betriebszuständen führen, wie zum Beispiel jahres- und tageszeitbedingte Schwankungen der Außentemperatur, Regen, Schnee oder starker Wind. Diese Einflüsse sollen nach Möglichkeit die Detektion der tatsächlichen kritischen Fehler nicht beeinflussen. Die untersuchten und entwickelten datenbasierten multivariaten Methoden sind mit dem Ziel der Entwicklung eines neuen Softwaremoduls mit den Möglichkeiten der Fehlerdetektion auf der Basis leistungsfähiger datenbasierter, statistischer Algorithmen in das Asset-Management-System zedas®asset der Firma PC-Soft GmbH zu integrieren. Dieses Modul soll die vorhandenen Diagnosemöglichkeiten von zedas®asset ergänzen. Da der gesamte Softwareentwicklungsprozess sowohl zeitlich als auch personell den Rahmen dieser Arbeit überschreitet, wird in diesem Rahmen lediglich die Architektur der Software entworfen und vorgestellt.

3 Grundlagen datenbasierter Softwareinformationssysteme zur Anlagenüberwachung

Zu Beginn des Kapitels erfolgt die Untersuchung und Beschreibung bekannter datenbasierter Methoden zur Fehlerdetektion. Der einleitende Abschnitt erläutert, welche Voraussetzungen die Prozessmessdaten erfüllen müssen, damit die in den datenbasierten Methoden enthaltenen statistischen Algorithmen erfolgreich zur Anwendung kommen können bzw. wie die Daten aufbereitet werden können, wenn diese Voraussetzungen so nicht gegeben sind. Die nachfolgenden Abschnitte beschreiben bekannte datenbasierte Methoden. Es werden die Algorithmen zum Anlernen des normalen oder fehlerfreien Zustandes und damit zur statistischen Modellbildung sowie die Algorithmen zur on-line Überwachung systematisch dargestellt und analysiert. Die vorgestellten Methoden bilden die Grundlage für das danach zu entwickelnde Softwareinformationssystem.

3.1 Motivation

Multivariate Methoden werden seit vielen Jahren zur Analyse großer Datenmengen insbesondere in der Ökonomie, Chemie und dem Versicherungswesen eingesetzt. Die multivariate Datenanalyse arbeitet aus der „Unmenge“ an Daten Informationen heraus, die danach mit weiteren analytischen Methoden bewertet werden müssen. Des Weiteren kann man über die Ermittlung von Zusammenhängen und Strukturen häufig Informationen über nicht direkt messbare Größen erhalten. Diese können ausgenutzt werden, um Schwachstellen im Herstellungsprozess eines Produktes festzustellen. Über diese indirekte Qualitätskontrolle kann man wiederum Einfluss auf den Produktionsprozess nehmen.

Zusammengefasst zählen laut Kessler [39] zu den Zielen der multivariaten Datenanalyse:

- Datenreduktion
- Vereinfachung
- Trennen von Informationen und Nicht-Informationen

- Datenmodellierung: Klassifizierung oder Regression
- Erkennen von Ausreißern
- Auswahl von Variablen
- Vorhersage
- Endmischen von Informationen.

Letztendlich können mithilfe der zum Anlernen benutzten Daten statistische Informationen über die zu untersuchenden Prozessen gebildet werden. Über die daraus erzeugten Parameter lässt sich das Normalverhalten eines Prozesses vom Fehlverhalten unterscheiden. In diesem Abschnitt sollen die am Häufigsten zur Fehlerdetektion und zum Prozessmonitoring verwendeten datenbasierten und statistischen Methoden betrachtet werden. Zunächst erfolgt eine Einführung in die grundlegenden Ideen, welche hinter diesen Methoden stecken. Nachfolgend wird näher auf die Kriterien für die Bestimmung der Schwellwerte eingegangen. Der Beschreibung der Standard-Methoden folgen die Vorstellungen abgewandelter Formen zur Überwachung nichtlinearer sowie dynamischer Prozesse.

3.2 Datenvorverarbeitung

3.2.1 Problemformulierung

Für die Anwendung multivariater Methoden, wie der Principal Component Analysis (PCA), ist die Vorverarbeitung von Daten ein wesentlicher Schwerpunkt, da sie auf der Grundannahme einer multivariaten Normalverteilung basiert. Fehlende Daten oder Ausreißer können zu erheblichem Informations- und Aussageverlust führen, weswegen die Statistik eine Reihe an Methoden bereitstellt, die es ermöglichen fehlende Werte zu ersetzen, beschrieben unter anderem in [46]. Auf einige Methoden wird nachfolgend näher eingegangen.

Die Aufbereitung erfolgt in drei Hauptschritten:

- Löschen von Variablen (Spalten), die zu wenige Daten enthalten
- Entfernen von Ausreißern
- Ersetzen von fehlenden Daten.

Weitere Manipulationen der Daten sind möglich, z.B. die Bildung von Ableitungen oder die Überführung der Daten in den Frequenzbereich. Die so berechneten Daten werden als neue Datenvektoren abgespeichert. Somit können in den Daten versteckte Informationen, die oft nicht sofort sichtbar sind, als zusätzliche Informationen für multivariate Verfahren herangezogen werden.

3.2.2 Behandlung von fehlenden Daten

Bei den in dieser Arbeit betrachteten industriellen Prozessen kann es vorkommen, dass Messdaten nur teilweise vorliegen, also Beobachtungen fehlen. Eine häufige Ursache für das plötzliche Fehlen von Daten ist der spontane Ausfall eines oder mehrerer Sensoren. Auch Probleme mit der Prozessleittechnik sowie Ausfälle der Kommunikationsinfrastruktur können zu Lücken in den Datenfolgen führen. Geeignete Fehlerdetektionsverfahren können Sensorausfälle feststellen und die betroffenen Variablen und dementsprechend auch die Sensoren benennen.

Eine weitere häufig verbreitete Ursache für fehlende Daten liegt in der Nutzung von Datenverdichtungsalgorithmen beim Speichern von Prozessdaten, z.B. dem sogenannten Toleranzbandverfahren. Das heißt, der neue Prozesswert wird nur gespeichert, wenn er um einen bestimmten Betrag oder um einen bestimmten Prozentsatz vom vorherigen Wert abweicht.

$$|y_i - y_{i-1}| < dev_{pre} \quad (3.1)$$

Dabei wird versucht, in den Daten vorhandene Redundanzen zu vermindern, wobei es ein gewisses Maß an Informationsverlust gibt. Ein nicht zu vernachlässigender Nebeneffekt dieser Art der Datenverdichtung ist, dass die Daten der Sensoren zu unterschiedlichen Zeiten ankommen, sodass sehr selten ein vollständiger Datenvektor über alle Sensoren zu einem bestimmten Zeitpunkt vorhanden ist.

Fehlende Daten können unter anderem mithilfe der Imputation ersetzt werden, wobei man nach [62] folgende Methoden unterscheidet:

- Mean Imputation
- Hot deck Imputation
- Cold deck Imputation
- Regression Imputation
- Multiple Imputation.

Einige ausgewählte Methoden werden nachfolgend näher betrachtet.

Fehlen nicht nur einzelne Samples, sondern größere Bereiche, so kann die Interpolation ein geeignetes Mittel sein, diese Daten zu erzeugen. Dazu werden die unbekannten Werte zwischen bekannten Datenpunkten gebildet. Des Weiteren können mit der Interpolation vollständige Datenvektoren mit einem einheitlichen Abtastintervall erzeugt werden.

Mean Imputation

Bei der Mean Imputation werden fehlende Daten durch den arithmetischen Mittelwert ersetzt. Zur Erhöhung der Robustheit, siehe „Behandlung von Ausreißern“, kann auch der Median zur Imputation herangezogen werden. Durch eine Mittelwertimputation erhöht sich zwar der Stichprobenumfang, nicht jedoch die Varianz, was zu einer Unterschätzung derselben führen kann [46].

Hot deck Imputation

Bei dieser Methode der Imputation werden die imputierten Werte aus einer geschätzten Verteilung ausgewählt. Meistens benutzt man hierfür die empirische Verteilung aus den beobachteten Werten. Problematisch ist es jedoch, wenn hinter dem Fehlen der Werte eine Systematik zu erkennen ist, z.B. eine höhere Fehlwahrscheinlichkeit bei großen Werten. In diesem Fall treten schnell Verzerrungen auf. Im Allgemeinen ist diese Methode aber sehr populär [46].

Cold deck Imputation

Hierbei wird ein konstanter Wert aus einer externen Quelle oder ein Erfahrungswert aus früheren Untersuchungen ersetzt. Das könnte beispielsweise ein Erfahrungswert aus dem Anlernen des Prozessregelverhaltens sein, wie es später unter dem „off-line Training“ beschrieben wird.

Regression Imputation

Viele Modellfunktionen $y(x)$ können als Linearkombination geeigneter Basisfunktionen $X_k(x)$ dargestellt werden:

$$y(x) = \sum_{k=1}^N a_k X_k(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2). \quad (3.2)$$

$\mathcal{N}(0, \sigma^2)$ heißt, dass die Daten standardnormalverteilt mit dem Mittelwert $\mu = 0$ und der Standardabweichung $\sigma = 1$ sind. Zur Bestimmung der einzelnen Parameter wird eine Regression innerhalb der vollständigen Fälle bestimmt und dann auf die unvollständigen Fälle angewandt. Wie in [62] beschrieben, können Unsicherheiten bei den Schätzungen der Regressionskoeffizienten bzw. Unsicherheiten bezüglich der Schätzung von σ^2 berücksichtigt werden. Interessant für den hier beschriebenen Fall der Fehlerdetektion mithilfe multivariater Verfahren ist die Schätzung der Unsicherheiten aus der Varianz $\hat{\sigma}^2$ über den Zusammenhang

$$(N - p) \frac{\hat{\sigma}^2}{\sigma^2} \sim \chi_{(N-p)}^2 \quad (3.3)$$

wobei p für die Anzahl der zu schätzenden Parameter steht. Dazu könnte man die Varianz aus dem „off-line Training“ verwenden.

Multiple Imputation

Bei der multiplen Imputation wird ein fehlender Wert durch $k \geq 2$ Werte ersetzt. Die k vervollständigten Datensätze können dann mit einer gewünschten Methode analysiert werden, sodass man dann k -Schätzungen für den interessierenden Wert erhält. Diese k -Schätzungen werden dann zu einer endgültigen Schätzung kombiniert, z.B. über eine Mittelwertbildung.

$$\hat{\theta} = \sum_{i=1}^k \frac{\hat{\theta}_i}{k} \quad (3.4)$$

Interpolation

Durch Interpolation kann der Verlauf einer kontinuierlichen Funktion zwischen bekannten Abtastpunkten (Stützstellen) unter der Annahme, dass der Verlauf weitestgehend „glatt“ ist, abgeschätzt werden. Von Vorteil ist, dass eine Interpolationskurve immer exakt durch alle Datenpunkte geht. Es gibt verschiedene Verfahren der Interpolation. Einige davon sind nachfolgend aufgeführt:

- Lineare Interpolation
- Nearest Neighbor Interpolation
- Spline Interpolation
- stückweise Hermite-Interpolation.

Die Nearest Neighbor Interpolation ist die einfachste Methode, um eine Funktion zu rekonstruieren. Für jede Position wird den Wert der nächstgelegenen Messstelle gewählt. Daraus ergibt sich eine stückweise konstante Funktion. Dieser Methode innewohnend liegt eine gewisse Fehlergröße, welche auch Auswirkungen im Frequenzbereich haben.

Da bei der Datenverdichtung, wie bereits beschrieben, lediglich ein sich ändernder Wert geschrieben wird, ist anzunehmen, dass der Prozesswert bis zu diesem Zeitpunkt nahezu die gleiche Größe hat. Deshalb wird die Nearest Neighbor Interpolation hier als das geeignete Interpolationsverfahren zur Datenrekonstruktion angesehen.

Es ist klar, dass kleine Schwankungen um den Prozesswert, welche bei der Datenverdichtung nicht berücksichtigt werden, auch nicht wiederhergestellt werden können. Darüber hinaus können höhere Frequenzanteile, die bei der Datenverdichtung verloren gegangen sind, auch durch anschließende Interpolation nicht wieder rekonstruiert werden.

3.2.3 Behandlung von Ausreißern

Ausreißer sind Proben mit atypischen Werten für diesen Parameter. Im statistischen Sinne bezeichnet man Ausreißer als Daten, die nicht zur Daten-Majorität gehören. Das Auftreten von Ausreißern in den Daten kann im Wesentlichen zwei Gründe haben: Einer davon ist das Auftreten eines experimentellen Fehlers, während der andere Grund das Eintreten eines einzigartigen Ereignisses sein kann. Es ist wichtig, beide Arten von Ausreißern zu identifizieren, da sie entweder von den Daten entfernt werden müssen, um korrekte Ergebnisse der Analyse zu erhalten, oder es müssen die Ursachen für das Auftreten der Ausreißer gefunden werden, um ein besseres Prozessverständnis zu erhalten. Somit können robustere Methoden gegen Ausreißer entwickelt werden. In der deskriptiven Statistik nutzt man oft zwei Ansätze zur Ausreißerbehandlung: Einerseits der Vergleich mit Lageparametern einer Häufigkeitsverteilung (Quantile) und andererseits die Bildung von sogenannten Z-Scores.

Quantile

Laut [24] ist ein Quantil folgendermaßen definiert: $P(X < x_p) \leq p$ und $P(X > x_p) \leq 1 - p$, wobei X eine Zufallsvariable, x_p ein bestimmtes p -Quantil und p eine Zahl zwischen 0 und 1 ist. Besondere Quantile sind bei $p = 0,5$ der Median und für $p = 0,25$ bzw. $p = 0,75$ das untere bzw. obere Quartil. Die Benutzung des Median für die Bestimmung der Datenmitte und davon ausgehend die Erkennung von Ausreißern erweist sich nach [13] als robustere Methode im Vergleich zur Benutzung des Mittelwertes. Für die grafische Veranschaulichung der Lage von Median, oberes und unteres Quartil wird oftmals der Boxplot benutzt. Nachfolgende Abbildung verdeutlicht den Zusammenhang zwischen Wahrscheinlichkeitsdichtefunktion und Boxplot.

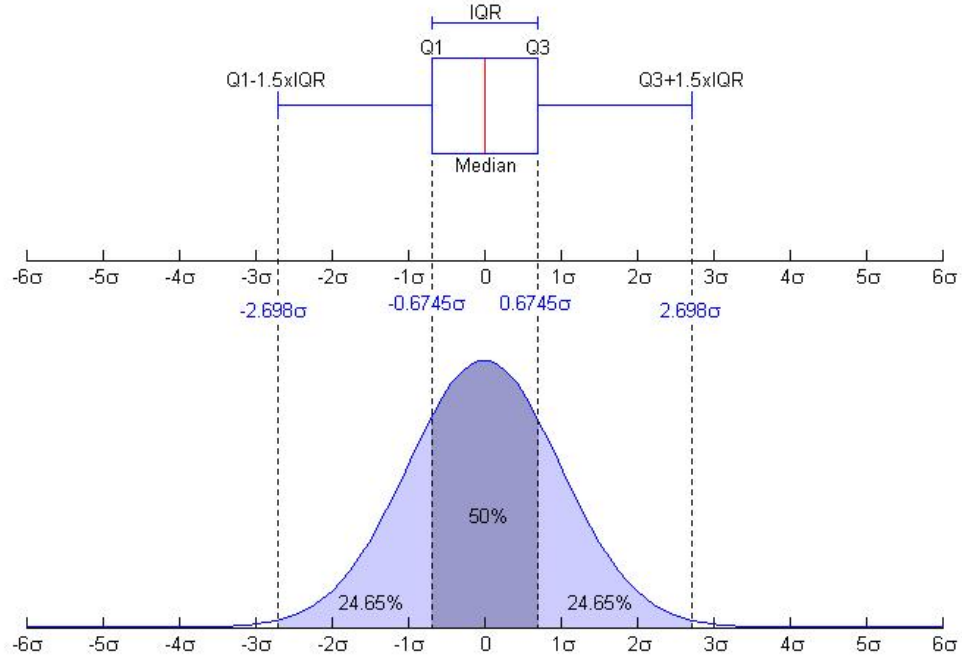


Abb. 3.1: Zusammenhang zwischen Boxplot und Dichtefunktion

Daten werden für gewöhnlich als Ausreißer erkannt und entfernt, wenn sie außerhalb des 1,5-fachen des Interquartilabstandes ausgehend vom oberen und unteren Quartil liegen. Befinden sich die Daten in multivariater Form, so erfolgt die Ausreißerbehandlung für jede Variable separat.

Z-Scores

Z-Scores werden bestimmt, indem man von jedem Sample einer Variable den Mittelwert abzieht und durch die Standardabweichung dividiert.

$$z = \frac{|x - \mu(x)|}{\sigma(x)} \quad (3.5)$$

Wie in [13] beschrieben wird zur Ausreißerbehandlung jeder Sample mit einem Z-Score größer 2,5 oder 3 entfernt. Zu beachten ist aber, dass zwischen 99,4% und 99,9% der Daten innerhalb dieses Intervalls liegen. Da auch hier der Einfluss auf die multivariaten statistischen Verfahren über Mittelwert und Standardabweichung recht groß ist, wird in [13] vorgeschlagen, die robuste Abschätzung von Mittelwert und Standardabweichung, $median(x)$ und σ_{Qn} zu benutzen.

$$z = \frac{|x - median(x)|}{\sigma_{Qn}(x)} \quad (3.6)$$

Laut [57] berechnet man σ_{Q_n} folgendermaßen

$$\sigma_{Q_n} = cf \cdot c \cdot \{|x_i - x_j; i < j|\}_{(k)}, \quad (3.7)$$

wobei cf ein Konsistenzfaktor, welcher von der Datengröße abhängt, $c = 2,2219$ ein konstanter Faktor und $k = \binom{h}{2} \approx \binom{n}{2}$ sind.

3.2.4 Zusammenfassung

Abschließend kann man festhalten, dass für die Fehlerdetektion unter der Problematik fehlender Daten eine Reihe an Möglichkeiten bereitsteht. Die Ergebnisse in [57] und [13] besagen, dass mit Imputationsmethoden gegenüber der Alternative, Daten bei Fehlen einzelner Informationen zu verwerfen, Verbesserungen erzielt werden können. Die Stärke dieser Verbesserungen hängt vom jeweiligen Sachverhalt ab, wobei es in bestimmten Situationen sogar vorkommen kann, dass überhaupt keine Verbesserung festzustellen ist. Laut [57] ist dies oftmals auf hohe Korrelationen, hohe Varianzen oder spezifische Problemstellungen zurückzuführen. Im Allgemeinen liefern aber die multiplen Imputationsmethoden und speziell die Mean Imputation recht gute und stabile Ergebnisse. Für die Erkennung von Ausreißern hat sich die Nutzung des 1,5-fachen des Interquartilabstandes als robuste Methode erwiesen. Eine Alternative dazu ist die Bildung der Z-Scores und die darauf basierende Ausreißererkennung. Hier sollte ebenfalls der Median sowie die robuste Abschätzung der Standardabweichung benutzt werden. Zur Rekonstruktion von Daten, bei denen Datenverdichtungsverfahren zur Anwendung kamen sowie zum Erzeugen vollständiger Datenvektoren mit gleichen Abstraten kann die Interpolation verwendet werden. Als für diesen Anwendungsfall geeignetes Verfahren hat sich die Nearest Neighbor Interpolation herausgestellt.

3.3 Datenbasierte Methoden der Fehlererkennung und Überwachung

3.3.1 Fehlererkennung auf Basis der PCA

Nachfolgend werden wesentliche Grundlagen zur Theorie der Hauptkomponentenanalyse erläutert. Die PCA ist eine grundlegende Methode im Bereich der multivariaten Statistik und Analyse. Sie fand bereits in verschiedensten Bereichen ihre Anwendung, wie z.B.:

- Datenkompression
- Mustererkennung
- Gesichtserkennung

- Signalanalyse und
- Prozessüberwachung [38].

Primäres Ziel ist die Entdeckung von Zusammenhängen zwischen Variablen, Objekten oder Merkmalen. Zu Beginn der Analyse liegt noch keine konkrete Vorstellung darüber vor, welche Beziehungszusammenhänge in den Datensätzen existieren. Es wird also versucht, aus den Variablen wenige latente, voneinander unabhängige Faktoren zu extrahieren. Die großen Vorteile dieses Verfahrens liegen darin, dass es analytisch fassbar ist, mit einer großen Anzahl von Prozessvariablen funktioniert und einen relativ geringen Rechenaufwand verursacht. Durch die Entwicklung von verschiedenen Varianten der PCA kam es in jüngster Zeit zu einem verstärkten Einsatz in der Prozessindustrie mit dem Fokus Fehlererkennung und Fehlerdiagnose.

Nachfolgend wird zunächst auf den Ansatz der Standard-PCA zur Fehlererkennung eingegangen. Der Einsatz der PCA erfolgt in der Regel in zwei Phasen, dem „off-line-Training“ oder auch „-Design“ und der „on-line-Überwachung“. Dieser Ansatz wurde für die Standard-PCA bereits in [32] beschrieben. Dafür werden Daten von Messwertgebern aufgenommen und in Spaltenvektoren (Beobachtungsvektoren) angeordnet. Die Anzahl der Messwertgeber und damit der Spalten ist gleich m .

3.3.1.1 Das off-line-Training (Design)

Für das „offline-Design“ werden N Datensätze des sogenannten „Regel- oder Normalbetriebes“ aufgenommen und im Wesentlichen in folgenden Schritten abgearbeitet:

- Normierung und Mittelzentrierung der Daten
- Berechnung der Singulärwerte und -vektoren
- Bestimmung der Hauptkomponenten
- Ermittlung der Schwellwerte.

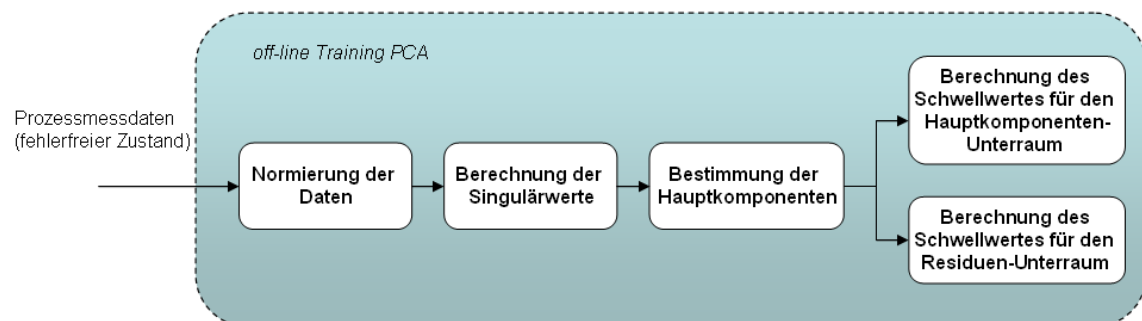


Abb. 3.2: Ablauf des PCA off-line Trainings

Normierung und Mittelzentrierung der Daten

Ausgangsbasis der Betrachtung ist die Datenmatrix $X \in R^{N \times m}$.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Nm} \end{bmatrix} \quad (3.8)$$

Im Allgemeinen werden die Daten nun bezüglich ihres Mittelwertes (Mittelwert = 0) und der Standardabweichung (Standardabweichung = 1) nach folgender Formel normalisiert.

$$z_i = \frac{x_i - \mu(X)}{\sigma(X)}, \quad (3.9)$$

wobei x_i der Zeilenvektor für eine Beobachtung aus X , $\mu(X)$ der Mittelwertvektor $\mu(X) = [\mu(x_1), \mu(x_2), \dots, \mu(x_m)]^T$ und $\sigma(X) = [\sigma(x_1), \sigma(x_2), \dots, \sigma(x_m)]^T$ sind. Der Mittelwert eines jeden Vektors berechnet sich über

$$\mu_j = \frac{1}{N} (x_{ij})^T 1_N, \forall j = [1 \dots m], i = [1 \dots N], \quad (3.10)$$

wobei $1_N = [1, 1, \dots, 1]^T \in R^N$ ist. Die Varianz als Quadrat der Standardabweichung ergibt sich für jede Messgröße $j = [1 \dots m]$ folgendermaßen:

$$Var_j(x_{ij}) = \sigma_j^2 = E[(x_{ij} - \mu_j)^T (x_{ij} - \mu_j)], \forall j = [1 \dots m], i = [1 \dots N]. \quad (3.11)$$

Im Ergebnis der Normalisierung erhält man die Datenmatrix

$$Z = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1m} \\ z_{21} & z_{22} & \dots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{N1} & z_{N2} & \dots & z_{Nm} \end{bmatrix} \in \mathcal{R}^{N \times m}. \quad (3.12)$$

Berechnung der Singulärwerte und Eigenvektoren der Kovarianzmatrix

Für die Berechnung der Eigenvektoren und Eigenwerte bzw. Singulärwerte unterscheidet man zwei grundlegende Verfahren,

- die Singulärwertzerlegung (SVD) und die
- Eigenwertzerlegung.

Zur Bestimmung der Eigenwerte sowie der -vektoren mit der Eigenwertzerlegung wird zunächst die Kovarianzmatrix

$$\Sigma \approx \frac{1}{N-1} Z^T Z \quad (3.13)$$

gebildet.

Diese hat folgende Eigenschaften [63]:

- Σ ist eine quadratische $m \times m$ Matrix.
- Die diagonalen Elemente von Σ sind die Varianzen der Messgrößen.
- Die nichtdiagonalen Elemente sind die Kovarianzen zwischen den verschiedenen Messgrößen.

Durch die Anwendung der Eigenwertzerlegung auf die Kovarianzmatrix erhält man letztendlich zwei Matrizen, P mit den Eigenvektoren und Λ mit den Eigenwerten,

$$\frac{1}{N-1} Z^T Z = P \Lambda P^T, \Lambda = \text{diag}(\sigma_1^2, \dots, \sigma_m^2) \quad (3.14)$$

wobei Λ eine diagonale Matrix ist, welche die Eigenwerte in absteigender Ordnung enthält: $\sigma_1^2 \geq \dots \geq \sigma_m^2 > 0$. Wird anstatt der Eigenwertzerlegung von Σ die Singulärwertzerlegung von Z benutzt, so entspricht das Quadrat des i^{ten} Singulärwertes dem i^{te} Eigenwert von Λ ($\sigma_i^2 = \lambda_i$). Ist aber die für die Singulärwertzerlegung verwendete Matrix, wie bei der Kovarianzmatrix quadratisch, symmetrisch und positiv definit, sind ihre Singulärwerte mit den Eigenwerten identisch. Des Weiteren haben unter diesen Bedingungen die linken und rechten Singulärvektoren die gleichen Werte und entsprechen den Eigenvektoren der Kovarianzmatrix.

Wenn die Eigenvektoren von solchen Merkmalen, welche die größten Varianzanteile besitzen, dominiert werden, ist es sinnvoll, die Singulärwertzerlegung auf der Basis der Korrelationsmatrix durchzuführen [27].

Geometrisch betrachtet rotieren die mithilfe der SVD gefundenen Eigenvektoren die Basis des m -dimensionalen Raumes. Die neuen Achsen sind die maximierten Varianzen, siehe Abbildung 3.3.

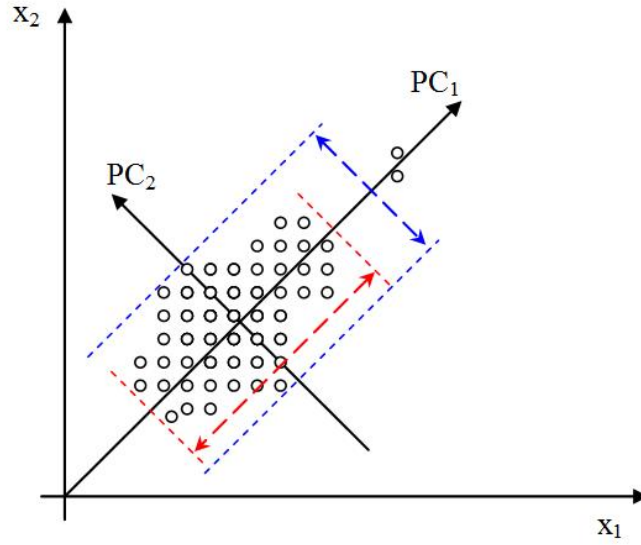


Abb. 3.3: Hauptachsentransformation für ein zweidimensionales Beispiel [39]

Bestimmung der Hauptkomponenten

Die Eigenvektoren mit den größten Eigenwerten spiegeln den größten Teil der Variationen wieder. Sie werden auch als Hauptkomponenten bezeichnet. Für die Bestimmung der Anzahl der Hauptkomponenten gibt es verschiedene Ansätze. Die Bekanntesten sind nachfolgend aufgeführt:

1. Kriterium von Kaiser: Alle Eigenwerte größer als der Mittelwert aller Eigenwerte zusammen bilden die Hauptkomponenten (siehe[33]).
2. Kriterium von Jolliffe: Jolliffe hat festgestellt, dass diese Forderung zu hoch ist und folglich alle Eigenwerte, die größer als das 0,7-fache des Mittelwertes aller Eigenwerte sind, zu den Hauptkomponenten gehören (siehe [26]).

Mithilfe der ermittelten Anzahl l der Hauptkomponenten kann der m -dimensionale Raum von P in einen Unterraum der Hauptkomponenten und in einen „Rest“ (engl. residual) -Unterraum aufgeteilt werden.

$$\Lambda = \begin{bmatrix} \Lambda_{pc} & 0 \\ 0 & \Lambda_{res} \end{bmatrix}, \Lambda_{pc} = \text{diag}(\sigma_1^2, \dots, \sigma_l^2) \in \mathcal{R}^{(l) \times (l)} \quad (3.15)$$

$$\Lambda_{res} = \text{diag}(\sigma_{l+1}^2, \dots, \sigma_m^2) \in \mathcal{R}^{(m-l) \times (m-l)}$$

$$P = \begin{bmatrix} P_{pc} & P_{res} \end{bmatrix} \in \mathcal{R}^{m \times m}, P_{pc} \in \mathcal{R}^{m \times l}, P_{res} \in \mathcal{R}^{m \times m-l} \quad (3.16)$$

wobei gilt, dass

$$\begin{bmatrix} P_{pc} & P_{res} \end{bmatrix} \begin{bmatrix} P_{pc} & P_{res} \end{bmatrix}^T = \begin{bmatrix} P_{pc} & P_{res} \end{bmatrix}^T \begin{bmatrix} P_{pc} & P_{res} \end{bmatrix} = I \quad (3.17)$$

ist. Die Untersuchung der Prozessdaten auf Abweichungen vom Normalzustand erfolgt in beiden Unterräumen. Dazu werden die original korrelierten Variablen in neue Variablen transformiert, die unkorreliert zueinander sind. Die lineare Transformation erfolgt nach:

$$T = Z \cdot P \quad (3.18)$$

In der Literatur wird die Matrix P mit den Eigenvektoren auch oft als „Loading“-Matrix bezeichnet, während man T „Score“-Matrix nennt. $T_{pc} = Z \cdot P_{pc}$ ist somit die Projektion der Original-Daten in den Unterraum der Hauptkomponenten, während $T_{res} = Z \cdot P_{res}$ die Projektion in den Residual-Unterraum ergibt.

Ermittlung der Schwellwerte

Für die Fehlererkennung werden im Wesentlichen zwei Kennwerte benutzt, die

- Hotelling's T^2 -Statistik (siehe [58]) sowie die
- SPE -Statistik (squared prediction error) (siehe [58]).

Der T^2 -Schwellwert

Der Schwellwert für die T^2 -Statistik kann zunächst über die multivariate Normalverteilung $\chi_l^2(\alpha)$ (multivariate Gauss-Verteilung) bestimmt werden. Mit ihr sind Wahrscheinlichkeiten als Anteile des Gesamtvolumens eines l -dimensionalen Ellipsoides (in normal verteilte Zufallsvariablen mit dem Erwartungswertvektor μ und der Kovarianzmatrix Σ) definiert.

Die auf das Volumen 1 normierte Funktion für die multivariate Normalverteilung lautet:

$$f(\underline{z}) = \frac{1}{(2\pi)^{\frac{l}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{z}-\underline{\mu})^T \Sigma^{-1}(\underline{z}-\underline{\mu})}. \quad (3.19)$$

Die in ihrem Argument auftretende Mahalanobis-Distanz erfüllt die Bedingung:

$$(\underline{z} - \underline{\mu})^T \Sigma^{-1}(\underline{z} - \underline{\mu}) \leq \chi_l^2(\alpha) \quad (3.20)$$

mit α einem zu setzenden alpha-Fehler-Niveau.

Alle Mahalanobis-Distanzen, die diese Bedingung erfüllen, erzeugen Konturen gleicher Wahrscheinlichkeit ($P = 1-\alpha$) in der multivariaten Normalverteilung.

Näherungsweise kann man den Schwellwert über die folgende Formel berechnen,

$$J_{th,T^2} = \frac{l(N-1)(N+1)}{N(N-l)} F_\alpha(l, N-l) \quad (3.21)$$

wobei

$$\frac{l(N-1)(N+1)}{N(N-l)} F_{\alpha}(l, n-l) \sim (\underline{z} - \underline{\mu})^T \Sigma^{-1} (\underline{z} - \underline{\mu}) \quad (3.22)$$

proportional der Mahalanobis-Distanz ist (siehe [51]). F_{α} ist die F-Verteilung, l die Anzahl der benutzten Hauptkomponenten und N die Anzahl der Datensätze.

Für ein vorgegebenes Niveau α ist der Schwellwert größer (konservativer) als $\chi_l^2(\alpha)$ [58].

Der SPE-Schwellwert

Die Verteilung im Residual-Unterraum, welche einer multivariaten Normalverteilung folgt, wurde durch Jackson und Mudholkar approximiert [32] und der nachfolgende Grenzwert abgeleitet

$$J_{th,SPE} = \theta_1 \left(\frac{c_{\alpha} \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{\frac{1}{h_0}} \quad (3.23)$$

mit

$$\theta_i = \sum_{j=l+1}^m (\sigma_j^2)^i, i = 1, 2, 3 \quad (3.24)$$

und

$$h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2}. \quad (3.25)$$

Die Variable l entspricht der Anzahl der Hauptkomponenten und c_{α} ist die normierte Abweichung entsprechend der oberen $1 - \alpha$ Perzentil.

Für die nachfolgende on-line-Überwachung benötigt man nun die Eigenvektoren P , die Singulär- oder auch Eigenwerte λ , die Anzahl der Hauptkomponenten l sowie die Mittelwerte μ und die Standardabweichungen σ der einzelnen Messgrößen des off-line-Designs.

3.3.1.2 Die on-line-Überwachung

Die „on-line-Überwachung“ erfolgt in den Schritten:

- Normierung und Mittelzentrierung der Daten
- Ermittlung der SPE- und T^2 -Indizes
- Auswertung der Indizes
- Prozessüberwachung und Fehlerdetektion.

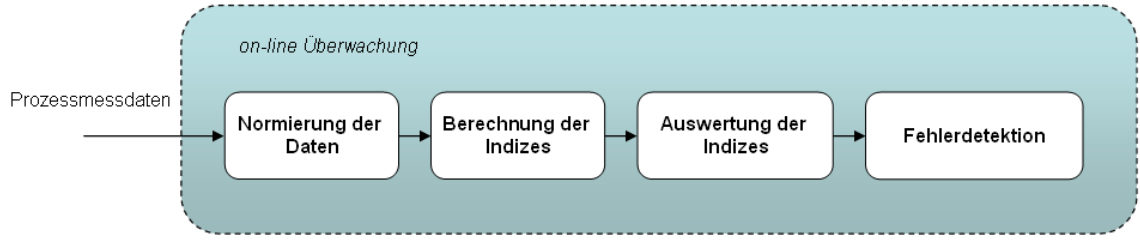


Abb. 3.4: Ablauf der on-line Überwachung

Normierung und Mittelzentrierung der Daten

Die Normierung und Mittelzentrierung der Daten erfolgt standardmäßig analog der Normierung und Mittelzentrierung des offline-Trainings. Der Mittelwert μ und die Standardabweichung σ werden vom off-line-Design für die Normierung herangezogen.

Ermittlung der SPE - und T^2 -Indizes

Für die on-line-Überwachung werden die Indizes entsprechend Hotelling's T^2 -Statistik und der SPE -Statistik ermittelt.

Jede der beiden Statistiken berechnet Indexwerte, die mit jeweils einem Grenzwert verglichen werden. Überschreitet einer der Indizes seinen Grenzwert, kann das System einen Alarm auslösen. Somit ist es möglich, die Daten in den beiden gebildeten Unterräumen zu überwachen.

Der T^2 -Index

Der T^2 -Index wird nach der folgenden Gleichung berechnet:

$$T^2 = z^T P_{pc} \Lambda_{pc}^{-1} P_{pc}^T z \quad (3.26)$$

Mit Worten ausgedrückt ist er die skalierte quadratische Norm eines Datenvektors z , ausgehend von seinem Mittelwert [58]. Die Skalierung ist indirekt proportional zur Standardabweichung in der Richtung der betrachteten Eigenvektoren.

Im zweidimensionalen Fall haben die auf die Hauptkomponenten projizierten Daten die Form einer Ellipse. Durch die Normierung mit den Varianzen erhält man für alle Punkte auf dem Ellipsenradius den gleichen Abstand C (Mahalanobis-Distanz - quadrierte Distanz im multivariaten Raum). Mit

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (3.27)$$

und

$$t = P^T z \quad (3.28)$$

erhält man eine Ellipsengleichung

$$t^T \Lambda^{-1} t \leq C \quad (3.29)$$

oder

$$\frac{t_1^2}{\lambda_1} + \frac{t_2^2}{\lambda_2} \leq C, \quad (3.30)$$

wobei $t = [t_1 \ t_2]^T$ ist.

Der Zusammenhang zwischen C und T^2 lautet: $T^2 = C$.

Legt man nun entsprechend der Anzahl der Hauptkomponenten l -Dimensionen zugrunde, erhält man die Gleichung (3.26).

Der SPE-Index

Der *SPE*-Index (squared prediction error), auch bekannt als Q-Statistik ist ein Messwert für die Projektion des Abtast-Datensatzes auf den Residual-Unterraum. Er ist das Quadrat der Entfernung des Abtastwertes rechtwinklig zum Unterraum der Hauptkomponenten.

$$SPE = \left\| \left(I - P_{pc} P_{pc}^T \right) z \right\|^2 = z^T \left(I - P_{pc} P_{pc}^T \right)^2 z \quad (3.31)$$

Prozessüberwachung und Fehlerdetektion

Die Fehlererkennung erfolgt nach der folgenden Logik

$$SPE \leq J_{th,SPE} \text{ or } T^2 \leq J_{th,T^2} \implies \text{fehlerfrei, ansonsten Fehler.}$$

Die *SPE*-Statistik misst folglich Fehler, die sich durch Veränderungen im Residuum - Unterraum bemerkbar machen, in welchem der Varianzanteil der Eigenvektoren nahe null liegt.

3.3.2 Die dynamische PCA - DPCA

Die klassische PCA liefert laut Mina und Verde [47] unter den folgenden Annahmen zuverlässige Ergebnisse:

- Die Daten folgen einer multivariaten Normalverteilung.
- Es existiert keine Autokorrelation innerhalb der Daten.
- Die Variablen behalten über die Beobachtungszeit ihren Mittelwert und die Standardabweichung bei.

Die Annahme bei der klassischen PCA, dass Messwerte zu einem Zeitpunkt statistisch unabhängig von Messwerten aus der Vergangenheit sind, trifft für viele Prozesse nicht zu. Der Status dynamischer Systeme zum Zeitpunkt t hängt von Messwerten zum Zeitpunkt $[t-h, t]$ ab. Dementsprechend sollten auch die gebildeten statistischen Modelle diese Abhängigkeiten berücksichtigen. Dynamische Systeme sind also mathematische Modelle für zeitabhängige Prozesse. Die dynamische PCA basiert auf der Idee, die Datenmatrix in der Form einer Hankel-Matrix aufzubauen, in deren von rechts oben nach links unten verlaufenden Nebendiagonalen jeweils die gleichen Werte vorkommen.

$$Z(h) = [Z(t)Z(t-1) \cdots Z(t-h)] \quad (3.32)$$

Folglich werden in die Matrix die gleichen aber zeitversetzten Vektoren integriert.

$$Z(h) = \begin{bmatrix} Z_t^T & Z_{t-1}^T & \cdots & Z_{t-h}^T \\ Z_{t-1}^T & Z_{t-2}^T & \cdots & Z_{t-h-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ Z_{t+h-N}^T & Z_{t+h-N-1}^T & \cdots & Z_{t-N}^T \end{bmatrix} \quad (3.33)$$

Die Kovarianzmatrix berechnet sich folgendermaßen:

$$\Sigma \approx \frac{1}{N-h-1} Z^T Z. \quad (3.34)$$

Dabei bezeichnet h die Anzahl der vorhergehenden Beobachtungen (eng. lags).

Wie die Anzahl der Lags automatisch bestimmt werden kann, ist in [42] beschrieben. Erfahrungen zeigen, dass für die Nutzung der DPCA für den Anwendungsfall „Prozessüberwachung“ die Wahl von $h = 1$ oder 2 ausreichend ist, entsprechend der Ordnung des dynamischen Systems. Bei nichtlinearen Systemen sollte die Anzahl der Lags größer gewählt werden, um eine bessere Approximation der Nichtlinearitäten zu erhalten [42].

Somit stellt die dynamische PCA eine Erweiterung der Standard-PCA dar. Die Phasen des off-line-Designs und der on-line-Überwachung erfolgen ansonsten analog den Phasen der Standard-PCA. Zu beachten ist, dass für die on-line-Berechnung der T^2 - und SPE -Indizes zum Zeitpunkt t der erweiterte Vektor $z_t(h)$ benötigt wird.

3.3.3 Rekursive PCA - RPCA

Das Verfahren des einmaligen Anlernens, wie es bei der klassischen PCA angewendet wird, hat Nachteile, wenn sich in dem Prozess über die Zeit Parameter ändern, ohne dass diese Parameteränderungen einen Fehler darstellen, sondern zum normalen Prozessverhalten gehören. Folglich kann es passieren, dass nichtkritische Änderungen im Prozessverhalten als Fehler interpretiert werden. Dem könnte man nun begegnen, indem man regelmäßig die Kovarianzmatrix neu ermittelt und nachfolgend die SVD

durchführt. Das ist aber sehr aufwendig und nimmt vergleichsweise viel Rechenzeit in Anspruch. Eine Alternative ist es, einen Algorithmus zu benutzen, der eine Erweiterung der klassischen PCA darstellt und auch als rekursive PCA (RPCA) bezeichnet wird. In dem nachfolgend beschriebenen Algorithmus werden wesentliche statistische Parameter, wie der Mittelwert und die Varianz, rekursiv auf der Basis neuer Messwerte berechnet. Zu seinen Vorteilen gehört unter anderem, dass er im Vergleich zu anderen rekursiven Algorithmen relativ wenig CPU-Rechenlast verursacht. Das Update der Eigenvektoren und der Eigenwerte erfolgt auf der Basis effizienter adaptiver Eigendekompositionstechniken. Wie bei der klassischen PCA unterscheidet man die Phasen des off-line-Designs und der on-line-Überwachung.

Das off-line-Training (Design)

Nach der Bildung der Kovarianzmatrix, der Eigenwertzerlegung und der Bestimmung der Anzahl der Hauptkomponenten, erhält man, wie bei der klassischen PCA, entsprechen den Gleichungen (3.15) und (3.16) nachfolgende Gleichungen:

$$\Lambda_0 = \begin{bmatrix} \Lambda_{pc_0} & 0 \\ 0 & \Lambda_{res_0} \end{bmatrix}, \Lambda_{pc_0} = \text{diag}(\sigma_1^2, \dots, \sigma_l^2) \in \mathcal{R}^{(l) \times (l)} \quad (3.35)$$

$$\Lambda_{res_0} = \text{diag}(\sigma_{l+1}^2, \dots, \sigma_m^2) \in \mathcal{R}^{(m-l) \times (m-l)}$$

$$P_0 = \begin{bmatrix} P_{pc_0} & P_{res_0} \end{bmatrix} \in \mathcal{R}^{m \times m}, P_{pc_0} \in \mathcal{R}^{m \times l}, P_{res_0} \in \mathcal{R}^{m \times m-l}. \quad (3.36)$$

Somit sind nach dem off-line-Design P_0, Λ_0 und l_0 die Anzahl der Hauptkomponenten bekannt.

Die on-line-Überwachung

Basierend auf den Gleichungen (3.15) und (3.16) erfolgt nun die rekursive Berechnung der Eigenvektoren und Eigenwerte. Das initiale PCA-Modell wird dazu mithilfe des neuen Datenvektors $z_k \in \mathcal{R}^m$ zum Zeitpunkt k aktualisiert.

$$\Lambda_k = \begin{bmatrix} \Lambda_{pc_{k-1}} + \Delta\Lambda_{pc_k} & 0 \\ 0 & \Lambda_{res_{k-1}} + \Delta\Lambda_{res_k} \end{bmatrix} \\ P_k = \begin{bmatrix} P_{pc_{k-1}} + \Delta P_{pc_k} & P_{res_{k-1}} + \Delta P_{res_k} \end{bmatrix} \quad (3.37)$$

Da die Kovarianzmatrix ein Produkt aus Eigenvektoren und Eigenwerten ist, kann die neue Eigenstruktur durch die Anpassung dieser Parameter auf Basis der neuen Werte z_k berechnet werden. Für den nachfolgend aufgeführten Algorithmus wird ein frei wählbarer Vergessensfaktor α eingeführt. Damit kann der zeitliche Faktor der Neubildung der Eigenvektoren und Eigenwerte gesteuert werden.

Zunächst wird der neue Datenvektor z_k mit dem Vergessensfaktor α belegt $z_k = \sqrt{1 - \alpha} z_k$ sowie der Faktor $f_i = p_{k-1,i}^T$ berechnet.

Die Anpassung der Eigenwerte $\lambda_{k,i}$ und Eigenvektoren $p_{k,i}$ erfolgt nach den Formeln

$$\begin{aligned}\lambda_{k,i} &= \alpha \lambda_{k-1,i} + f_i^2 \\ p_{k,i} &= p_{k-1,i} + \sum_{j=1}^m b_{ij} p_{k-1,j} \\ b_{ij} &= \begin{cases} 0 & j = i \\ -b_{ij} = \frac{f_i f_j}{(\lambda_{k-1,i} - \lambda_{k-1,j})} & j \neq i \end{cases}\end{aligned}\quad (3.38)$$

Der Algorithmus zur rekursiven Berechnung ist in [19] beschrieben.

3.3.4 Kernel-PCA

Kernel-Theorie

Die bisher vorgestellten Ansätze der multivariaten Verfahren, wie zum Beispiel die Hauptkomponentenanalyse, gehen davon aus, dass sich die Strukturen der Daten im linearen Raum gut beschreiben bzw. die Daten sich linear voneinander trennen lassen. Bei nichtlinearen Prozessen ist das oft nicht mehr gegeben. Transformiert man die Daten aber in einen höher dimensional nichtlinearen Raum $z \in Z \rightarrow H$, ist eine lineare Trennung der Daten möglich.

Da die Funktion (3.13) der PCA vom Skalarprodukt $\langle z^T, z^* \rangle$ abhängt, kann man eine Kernelfunktion zur Projektion $z_i \rightarrow \Phi(z_i)$ in einen höher dimensional Raum einsetzen. Mithilfe der sogenannten Kernelfunktionen ist es möglich, auch nichtlineare Zusammenhänge mit linearen Methoden zu behandeln. Da die Berechnung von den Projektionen $\Phi(z_i)$ sehr rechenintensiv (oder unmöglich) sein kann, kommt der sogenannte Kerneltrick zum Einsatz. Dabei müssen die Projektionen der Eingabevektoren $\Phi(z_i)$ nicht explizit berechnet werden, es genügt die Kernelfunktion $k(z_i, z_j) = \Phi(z_i) \cdot \Phi(z_j) = \langle \Phi(z_i), \Phi(z_j) \rangle$ zu kennen. Mit dieser nichtlinearen Funktion kann das Skalarprodukt direkt im Eingaberaum berechnet werden.

Es gibt verschiedene Arten von Kernel, z.B.:

- Lineare Kernel: $k(z_i, z_j) = \langle z_i, z_j \rangle$
- Polynomielle Kernel: $k(z_i, z_j) = \langle z_i, z_j \rangle^d$
- RBF-Kernel (Radial-Basisfunktionen-Kernel): $k(z_i, z_j) = \exp\left(-\frac{\|z_i - z_j\|^2}{2\sigma^2}\right)$.

Anwendung von Kernel in der PCA

Die Kovarianzmatrix wird nun nicht mit den Daten des nichtlinearen Eingaberaumes gebildet, sondern aus den transformierten Daten in einem linearen Merkmalsraum H

$$\Sigma^H = \frac{1}{N} \sum_{j=1}^N \Phi(z_j) \Phi(z_j)^T, \quad (3.39)$$

wobei $\Phi(\cdot)$ eine nichtlineare Mapping Funktion ist, welche den Input-Vektor in den nichtlinearen Raum H projiziert.

Analog zu Gleichung (3.18) kann die neu erhaltene Kovarianzmatrix in Eigenvektoren und Eigenwerte zerlegt werden. Wie bei der normalen PCA enthalten die Eigenvektoren mit den größten Eigenwerten nach der Singulärwertzerlegung den größten Informationsanteil (Varianz).

$$\Sigma^H = P \Lambda P^T. \quad (3.40)$$

Die Projektionen der Messwerte auf die Hauptkomponenten werden folgendermaßen berechnet:

$$t_k = \langle p_k, \Phi(z) \rangle = \sum_{i=1}^N \alpha_i^k \langle \Phi(z_i), \Phi(z) \rangle. \quad (3.41)$$

p_k sind die k Eigenvektoren mit dem größten Varianzanteil im Merkmalsraum.

Über den bereits beschriebenen Zusammenhang

$k(z_i, z_j) = \Phi(z_i) \cdot \Phi(z_j) = \langle \Phi(z_i), \Phi(z_j) \rangle$ erhält man

$$t_k = \sum_{i=1}^N \alpha_i^k K(z_i, z). \quad (3.42)$$

Die Koeffizienten α^k stellen die orthonormalen Eigenvektoren von K dar und sind so zu normalisieren, dass $\langle \alpha_k, \alpha_k \rangle = \frac{1}{\lambda_k}$ ergibt.

Vor der Bestimmung der Eigenwerte und Eigenvektoren sollte K zentralisiert werden. Das ist im Merkmalsraum H nicht so einfach, wie im Eingaberaum. In [60] wurde deshalb folgendes Verfahren vorgeschlagen:

$$\tilde{K} = K - I_N K - K I_N + I_N K I_N. \quad (3.43)$$

I_N ergibt sich über

$$I_N = \frac{1}{N} \begin{pmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{pmatrix} \in R^{N \times N}. \quad (3.44)$$

Die Kernelmatrix K aus der Gleichung (3.42) ist einfach durch \tilde{K} zu ersetzen.

Für die on-line-Überwachung (Bildung der Indexe T^2 und SPE) kommt zur Zentralisierung der Daten eine etwas abgewandelte Form der Gleichung (3.43) zum Einsatz.

$$\tilde{k}_t = k_t - I_t K - k_t I_N + I_t K I_N . \quad (3.45)$$

k_t ist die Kernelmatrix der Testdaten.

Den T^2 - Index bildet man mithilfe der Gleichung (3.42) und erhält

$$T^2 = [t_1, \dots, t_k] \Lambda^{-1} [t_1, \dots, t_k]^T . \quad (3.46)$$

Der Grenzwert für die T^2 Statistik wird, wie gehabt, mit der F-Statistik über die Gleichung (3.21) berechnet.

Den SPE-Index berechnet man laut Gleichung (3.31) über den nachfolgenden Zusammenhang:

$$SPE = \|(I - PP^T)z\|^2 = \|\Phi(z) - \hat{\Phi}_p(z)\|^2 , \quad (3.47)$$

wobei $\hat{\Phi}_p(z) = \sum_{k=1}^p t_k p_k$ für die p Hauptkomponenten ist, während $\Phi(z) = \sum_{k=1}^n t_k p_k$ für alle Eigenvektoren, bei denen der Eigenwert ungleich null ist.

Entsprechend der Herleitung in [43] ergibt sich für den SPE-Index die folgende Gleichung:

$$SPE = \sum_{j=1}^n t_j^2 - \sum_{j=1}^p t_j^2 . \quad (3.48)$$

Der SPE-Schwellwert kann über den Zusammenhang $SPE_\alpha \sim g\chi_h$ ermittelt werden, wobei $g = \frac{b}{2a}$ und $h = \frac{2a^2}{b}$ sind. Die Variable a ist hier der erwartete Mittelwert und b die Varianz des SPE-Indexes.

3.3.5 Änderungsdetektion mit dem Generalized Likelihood Ratio (GLR)

Inhalt dieses Abschnittes ist die Beschreibung der statistischen Methode des Likelihood Ratio (LR), welche man ebenfalls zur Fehlerdetektion verwenden kann. Es wird gezeigt, dass man für die Fehlerdetektion die typischen Schritte, wie Residuengenerierung und -auswertung, das Ermitteln der Schwellwerte sowie die Entscheidung Fehler oder kein Fehler abarbeiten muss. Ausgangspunkt ist die Methode der maximalen Stichprobenwahrscheinlichkeit (engl. maximum likelihood-method). Bei dieser Methode geht man im Allgemeinen davon aus, dass die Art der Verteilung bekannt ist und nur die konkreten Parameter der bekannten Verteilung ermittelt werden müssen. Im Fall der vorliegenden Normalverteilung wären das z.B. der Erwartungswert

und die Varianz. Im Rahmen der Fehlerdetektion ist es das Ziel, aus den beobachteten Werten (bekannten Verteilungen) die Änderung eines Modellparameters θ abzuschätzen, sodass ein optimaler Schwellwert für die Detektion von Fehlern bestimmt werden kann. Der gesuchte Parameter wird so gewählt, dass die erhaltene Stichprobe die größte Auftretswahrscheinlichkeit hat. Deshalb erfolgt zunächst eine Erklärung für den Maximum-Likelihood-Schätzer.

Maximum-Likelihood-Schätzer

X_1, \dots, X_m sind unabhängige Stichprobenvariablen, die verteilt sind wie eine Zufallsvariable \mathbf{X} mit der Dichtefunktion $p^{\mathbf{X}}(\mathbf{x}; \theta)$. Die Beobachtungen $\mathbf{x}_1, \dots, \mathbf{x}_N$ sind unabhängig und stammen aus der $\mathcal{N}_m(\mu, \Sigma)$ -verteilten Grundgesamtheit. Die gemeinsame Dichtefunktion von X_1, \dots, X_m ist laut Ross in [55] gegeben durch

$$p^{\mathbf{X}}(\mathbf{x}_1, \dots, \mathbf{x}_N; \theta) = p^{\mathbf{X}}(\mathbf{x}_1; \theta) \cdot \dots \cdot p^{\mathbf{X}}(\mathbf{x}_N; \theta) = \prod_{i=1}^N p^{\mathbf{X}}(\mathbf{x}_i; \theta). \quad (3.49)$$

Laut Ross [55] ist nun die plausibelste Schätzung für θ (die Maximum-Likelihood-Schätzung) der Wert, bei dem die Wahrscheinlichkeit des Auftretens der beobachteten Werte maximal ist ($\hat{\theta}$).

$$L(\hat{\theta}; \mathbf{X}) \geq L(\theta; \mathbf{X}) = \prod_{i=1}^N p^{\mathbf{X}}(\mathbf{x}_i; \theta) \quad \forall \theta \in \Theta. \quad (3.50)$$

Des Weiteren wird ausgenutzt, dass $p^{\mathbf{X}}(\mathbf{x}_1, \dots, \mathbf{x}_N; \theta)$ und $\ln[p^{\mathbf{X}}(\mathbf{x}_1, \dots, \mathbf{x}_N; \theta)]$ ihr Maximum bei dem selben Wert von θ haben.

$$\ln(L(\hat{\theta}; \mathbf{X})) = l(\hat{\theta}; \mathbf{X}) \quad (3.51)$$

$$l(\hat{\theta}; \mathbf{X}) \geq l(\theta; \mathbf{X}) = \sum_{i=1}^N \ln p^{\mathbf{X}}(\mathbf{x}_i; \theta) \quad \forall \theta \in \Theta. \quad (3.52)$$

Während $L(\theta; \mathbf{X})$ die Likelihood-Funktion von θ ist, stellt $l(\theta; \mathbf{X})$ die Log-Likelihood-Funktion von θ dar. Der Maximum-Likelihood-Schätzer (ML-Schätzer) wird bestimmt, indem man die erste Ableitung der Likelihood-Funktion berechnet und sie mit null gleichsetzt. Das Maximum liegt vor, wenn die zweite Ableitung der Likelihood-Funktion kleiner als null ist.

Für die multivariate Normalverteilung hat die Likelihood-Funktion im Allgemeinen die Form:

$$L(\mu, \Sigma; \mathbf{X}) = (2\pi)^{-\frac{Nm}{2}} |\Sigma|^{-\frac{N}{2}} \exp \left\{ -\frac{1}{2} \sum_{i=1}^N (\mathbf{x}_i - \mu)^T \Sigma^{-1} (\mathbf{x}_i - \mu) \right\}. \quad (3.53)$$

Die Log-Likelihood-Funktion für die multivariate Normalverteilung sieht folgendermaßen aus:

$$l(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{X}) = \ln(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{X}) = -\frac{Nm}{2} \ln(2\pi) - \frac{N}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}). \quad (3.54)$$

Durch partielles Ableiten von 3.54 und Nullsetzen der partiellen Ableitungen erhält man die ML-Schätzer für $\boldsymbol{\mu}$ und $\boldsymbol{\Sigma}$.

$$\hat{\boldsymbol{\mu}} = \bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i \quad (3.55)$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad (3.56)$$

Testen von Hypothesen

Um mithilfe statistischer Methoden eine Entscheidung treffen zu können, ob ein Fehler vorliegt, bedient man sich der Hypothesentests. Laut Ross [55] ist eine Hypothese eine Aussage über einen oder mehrere Parameter, die in der Verteilung einer Gesamtheit eine Rolle spielen. Ein Test einer Hypothese ist eine Vorschrift, ob die Hypothese zurückgewiesen werden soll oder nicht. Hat man eine Grundgesamtheit mit einer gegebenen Verteilung, kann man nun eine Hypothese H_0 gleich Nullhypothese zu einem unbekannten Parameter θ aufstellen und testen. Die Hypothese H_1 ist die Alternativhypothese zu H_0 . Beide Hypothesen schließen sich bezüglich der interessierenden Größen gegenseitig aus. Mithilfe des statistischen Tests soll eine Entscheidung zwischen der Null- und der Alternativhypothese getroffen werden.

Beim Testen können nun zwei verschiedene Arten von Fehlern auftreten. Bei dem sogenannten Fehler 1. Art wurde die Nullhypothese fälschlicherweise abgelehnt, während bei dem Fehler 2. Art die Nullhypothese fälschlicherweise angenommen wurde.

Die aus Bourrier [5] entnommene Tabelle stellt die möglichen Kombinationen übersichtlich dar.

Entscheidung/ wahrer Zustand	H_0	H_1
Beibehaltung von H_0	richtige Entscheidung	Fehler 2. Art ($1 - \alpha$)
Ablehnung von H_0	Fehler 1. Art (α)	richtige Entscheidung

Tabelle 3.1: Kombinationen - wahrer Zustand, Entscheidung und Folgen

Likelihood-Quotiententest

Einer der bekanntesten mathematischen Hypothesentests, der hier im Rahmen der Fehlerdetektion zur Anwendung kommt, ist der Likelihood-Quotienten-Test. Nach diesem Prinzip entscheidet man sich für die Alternativhypothese, wenn die Dichtefunktion $p_{\theta_1}(x)$ hinreichend stark über $p_{\theta_0}(x)$ dominiert.

Die Größe des Likelihood-Quotienten nach dem Lemma von Neyman-Pearson, beschrieben unter anderem in [25], ist Ausdruck für den Grad der Dominanz von $p_{\theta_1}(x)$ über $p_{\theta_0}(x)$.

$$LQ = \begin{cases} \frac{p_{\theta_1}(x)}{p_{\theta_0}(x)} & \text{falls } p_{\theta_0}(x) > 0 \\ \infty & \text{falls } p_{\theta_0}(x) = 0 < p_{\theta_1}(x) \end{cases} \quad (3.57)$$

Hinreichend starke Dominanz bedeutet, dass der Likelihood-Quotient den festgelegten Schwellenwert c übersteigt (siehe Abbildung 3.5).

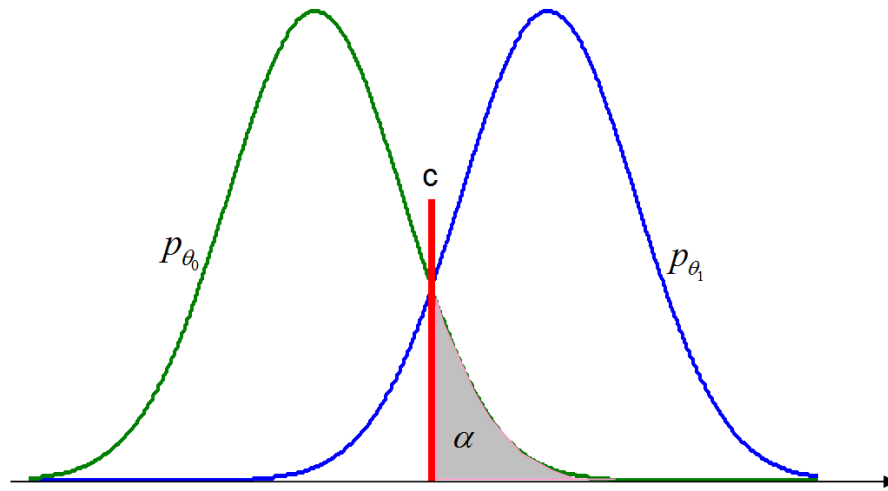


Abb. 3.5: Konstruktion eines Neyman-Pearson Tests

In der Literatur findet man oftmals, dass der Likelihood-Quotient genau umgekehrt angegeben wird, also der Wert der Likelihood-Funktion unter H_0 dividiert durch den Wert der Likelihood-Funktion unter H_1 (siehe [22]).

$$\lambda = \frac{p_{\theta_0}(x)}{p_{\theta_1}(x)} \quad (3.58)$$

Folglich wird H_0 abgelehnt, falls $\lambda < \lambda_0$ ist. λ_0 ist der gewählte Schwellenwert bei dem Signifikanzniveau α . Da $\theta_0 \subset \theta$ ist, kann λ Werte zwischen $0 \leq \lambda \leq 1$ annehmen.

In ihrer Abhandlung [2] verwendeten Basseville und Nikiforov für den Likelihood-Quotient anstatt der Likelihood-Funktionen die logarithmierten Likelihood-Funktionen

(siehe Abschnitt 3.3.5). Der logarithmierte Likelihood-Quotient (Log-Likelihood-Quotient) für einen Wert x_i ist laut [2] folgendermaßen definiert:

$$lq(x_i) = \ln \frac{p_{\theta_1}(x_i)}{p_{\theta_0}(x_i)}, \quad (3.59)$$

wobei $p_{\theta}(x_i)$ die Wahrscheinlichkeitsdichte für $x = x_i$ ist. Für eine feste Fenstergröße von N Werten ergibt sich:

$$lq_1^N = \sum_1^N s_i, \quad (3.60)$$

Generalized-Likelihood-Quotiententest

Das für die Testtheorie grundlegende Lemma von Neyman-Pearson zeigt, dass eine solche Testkonstruktion zu einem optimalen Testverfahren führen kann. Im Fall der zwei einfachen zusammengesetzten Hypothesen H_0 und H_1 lässt sich die Gleichung 3.57 erweitern, in dem man bei der Nullhypothese und der alternativen Hypothese jeweils die höchste Likelihood bestimmt und diese beiden ins Verhältnis setzt.

$$LQ(x) = \frac{\sup_{\theta \in \Theta_1} \prod_{i=1}^N p^X(x_i; \theta)}{\sup_{\theta \in \Theta_0} \prod_{i=1}^N p^X(x_i; \theta)} \quad (3.61)$$

Der Zähler enthält als Grundmenge den Raum der Nullhypothese, also Θ_1 , während für den Nenner die Grundmenge Θ_0 betrachtet wird. Erreicht $LQ(x)$ einen Wert nahe eins bedeutet dies, dass aus den vorhandenen Daten ein Unterschied zwischen den beiden Paramettermengen Θ_1 und Θ_0 zu erkennen ist. Die Nullhypothese wird in diesen Fällen abgelehnt. Umgekehrt wird bei einem Likelihood-Quotienten-Test die Hypothese H_0 bei einem Signifikanzniveau α nicht abgelehnt, falls

$$LQ(x) < c_{\alpha} . \quad (3.62)$$

c stellt den kritischen Wert laut siehe Abbildung 3.5 dar. Er ist so zu wählen, dass gilt:

$$\sup_{\theta \in \Theta_0} P_{\theta}(LQ(x) < c_{\alpha}) = \alpha. \quad (3.63)$$

Wie in [14] beschrieben, ist in der Praxis oftmals θ_1 nicht bekannt. Um dennoch Änderungen von θ bei Unbekanntem θ_1 zu detektieren, wurde die Verwendung der von Basseville und Nikiforov in [2] vorgestellten Methode des Generalized-Likelihood-Quotienten (engl. Generalized-Likelihood-Ratio GLR) vorgeschlagen. Basierend auf dem Log-Likelihood-Quotienten für bekannte θ_1 laut Gleichung 3.59 wird für unbekannte θ_1 dieses durch den Maximum-Likelihood-Schätzer ersetzt. Die Maximum-Likelihood-Schätzung von θ_1 kann nach Basseville und Nikiforov in [2] durch die Maximierung von LQ erreicht werden.

Der Generalized-Likelihood-Quotient g berechnet sich nun folgendermaßen:

$$g(x) = \ln \hat{L}Q(x) = \sup_{\theta_1} lq(x) \quad (3.64)$$

Der auf dem GLR beruhende Generalized-Likelihood-Quotiententest testet zwischen den Hypothesen $H_0 = \{\theta : \theta \in \Theta_0\}$ und $H_1 = \{\theta : \theta \in \Theta_1\}$. Dabei ist

$$\hat{g}(x) = \begin{cases} 1 & \text{wenn } \hat{L}Q(x) \geq c_\alpha \\ 0 & \text{wenn } \hat{L}Q(x) < c_\alpha, \end{cases} \quad (3.65)$$

wobei der kritische Wert c_α so gewählt wird, dass

$$\sup_{\theta \in \Theta_0} P_\theta(\hat{L}Q(x) < c_\alpha) = \alpha \quad (3.66)$$

ist.

Für den Fall, das $H_0 : \mu = \mu_0$, $H_1 : \mu \neq \mu_0$ und die Kovarianzmatrix bekannt sind, N Samples zur Verfügung stehen und \bar{x} der Schätzer für μ aus Gleichung 3.55 ist, erhält man in der Vektor-Schreibweise über die Gleichungen 3.59, 3.53 die Gleichung

$$lq(x) = \ln \frac{p_{\theta_1}(x)}{p_{\theta_0}(x)} = -\frac{1}{2} \left[\sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) - \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}) \right] \quad (3.67)$$

und letztendlich die Teststatistik

$$-2lq(x) = N \left[(\bar{\mathbf{x}} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\bar{\mathbf{x}} - \boldsymbol{\mu}) \right]. \quad (3.68)$$

Werden für $\boldsymbol{\Sigma}$ die normierte Kovarianzmatrix \mathbf{S} und für \mathbf{x} die normierten Werte \mathbf{z} laut Gleichung 3.9 eingesetzt, so ergibt sich für die Teststatistik:

$$-2lq(x) = N \bar{\mathbf{z}}^T \mathbf{S}^{-1} \bar{\mathbf{z}}. \quad (3.69)$$

Wie in dem Buch von Farneir, Hamerle und Tutz [22] beschrieben, ist

$$N \bar{\mathbf{z}}^T \mathbf{S}^{-1} \bar{\mathbf{z}} \sim T^2(m, N) \quad (3.70)$$

und sowohl für $m = 1$ als auch für $m > 1$ ist die T^2 -Verteilung der F -Verteilung äquivalent. $T^2(m, N)$ ist Hotelling's T^2 -Verteilung, unter anderem beschrieben in [22].

$$\frac{N - m}{(N - 1)m} T^2 \sim F(m, N - m) \quad (3.71)$$

Laut Farneir, Hamerle und Tutz [22] ergibt sich damit folgende Testvorschrift:

H_0 wird abgelehnt, falls $\frac{(N-m)N}{(N-1)m} C > F(m, N - m; 1 - \alpha)$ ist.

C ist die bereits in Abschnitt 3.3.1.1 beschriebene Mahalanobis-Distanz.

Folglich erhält man für den Vertrauensbeweis $(1 - \alpha)$

$$C \leq \frac{(N-1)m}{(N-m)N} F(m, N-m; 1-\alpha). \quad (3.72)$$

Das off-line-Training (Design)

Wie bei der PCA bildet man mithilfe des Trainingsdatensatzes die normierte Kovarianzmatrix \mathbf{S}^{-1} sowie Mittelwert $\boldsymbol{\mu}$ und Standardabweichung $\boldsymbol{\sigma}$. Der Schwellwert J_{Th} wird über die F-Statistik laut Gleichung 3.72 errechnet.

Nachfolgende Abbildung beschreibt die wesentlichen Schritte für das off-line Training:

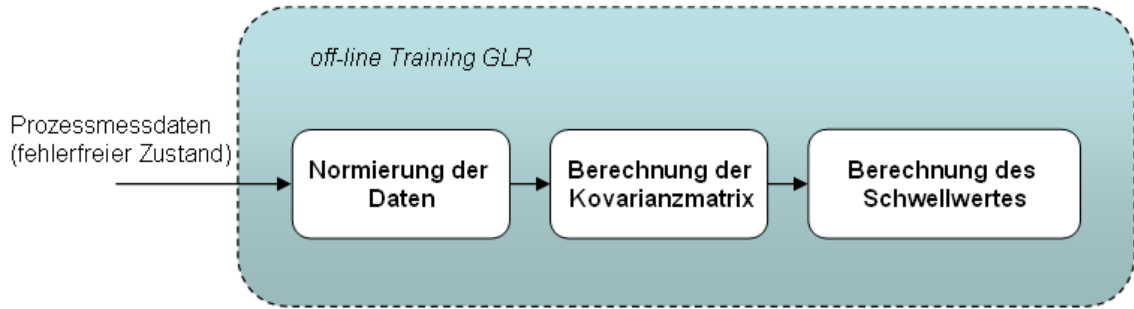


Abb. 3.6: Ablauf des GLR off-line Trainings

Die on-line-Überwachung

Die on-line-Überwachung erfolgt im Wesentlichen in den gleichen Schritten wie bei der PCA, wobei statt der Eigenwerte nun die gesamte Kovarianzmatrix des off-line-Trainings zur Anwendung kommt.

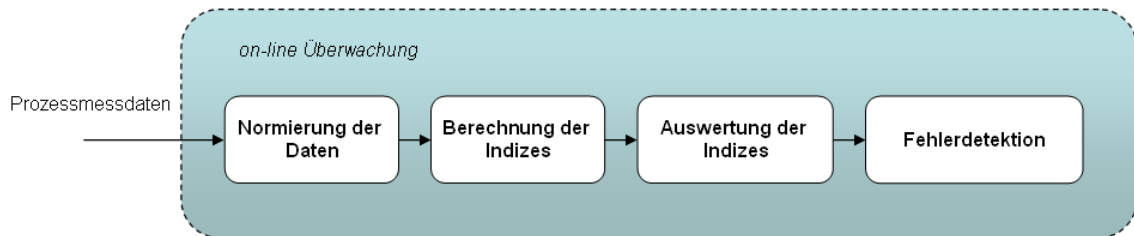


Abb. 3.7: Ablauf der on-line Überwachung

Die Logik für die Überwachung kann man folgendermaßen beschreiben:

$$N\bar{\mathbf{z}}^T \mathbf{S}^{-1} \bar{\mathbf{z}} \leq J_{Th}. \quad (3.73)$$

Der Prozess wird als fehlerfrei betrachtet, solange diese Bedingung erfüllt ist.

3.3.6 Zusammenfassung

Die wesentlichen Ergebnisse dieses Abschnittes sind:

- Die bekanntesten zur Fehlererkennung und zum Prozessmonitoring verwendeten datenbasierten und statistischen Methoden wurden untersucht. Es konnte gezeigt werden, dass diese Methoden für die Fehlerdetektion mit multivariaten Daten geeignet sind. Unter der Zielstellung ein Überwachungssystem auf der Basis multivariater Verfahren zu realisieren, wurden fünf Methoden zur Fehlerdetektion vorgestellt, wobei drei davon Abwandlungen oder Erweiterungen der Hauptkomponentenanalyse darstellen.
- Die Standard-PCA ist sicherlich eine der am häufigsten angewendeten datenbasierten Methoden zur Fehlerdetektion, obwohl sie ursprünglich für andere Anwendungsfälle entwickelt wurde. Alle Methoden einigt, dass immer in zwei Phasen vorgegangen wird, dem off-line-Training und der on-line-Überwachung. In dem off-line-Training werden die wichtigsten Parameter für das „statistische Modell“ sowie die Grenzwerte auf der Basis bekannter multivariater Statistiken ermittelt.
- Beim Auftreten von seriellen Korrelationen in den Daten ist die dynamische PCA besser zur Fehlerdetektion geeignet als die Standard-PCA. Die Datenmatrix muss dazu in die Form einer Hankel-Matrix gebracht werden.
- Bei schleichenden Veränderungen von Messwerten, die aber kein Ausdruck für Fehler sind, versagt die Standard-PCA. Ein dauerhaftes Überschreiten der Grenzwerte ist die Folge. Abhilfe schafft hier die rekursive Erneuerung wesentlicher Modellparameter, wie Eigenwerte und Eigenvektoren. Die dazu notwendigen Algorithmen wurden vorgestellt und erläutert.
- Treten Nichtlinearitäten im Prozess auf, liefert die Standard-PCA ebenfalls keine befriedigenden Ergebnisse. Mithilfe von Kernelfunktionen und dem „Kerneltrick“ können auch nichtlineare Zusammenhänge mit linearen Methoden behandelt werden. Zu beachten ist, dass der Rechenaufwand durch die Verwendung der Kernelmatrix deutlich steigt.
- Eine bisher wenig beachtete Methode zur Fehlerdetektion an industriellen Prozessen ist die Methode des Generalized-Likelihood-Quotienten. Die Grenzwerte werden hier mithilfe des Generalized-Likelihood-Quotienten unter Verwendung des Maximum-Likelihood-Schätzer berechnet.

Diese Ergebnisse stellen eine wesentliche Grundlage für den Entwurf eines datenbasierten Fehlerdiagnosesystems für Gurtförderanlagen dar. Die praktische Anwendbarkeit und Integrierbarkeit in ein Softwareprodukt ist anhand dieser industriellen Aufgabenstellung zu untersuchen.

3.4 Anforderungen und Richtlinien an ein neues datenbasiertes Softwarediagnose und -informationssystem

Inhalt dieses Abschnitts ist die Beschreibung der Anforderungen sowie die Einordnung der datenbasierten statistischen Verfahren zur Fehlererkennung in die Struktur eines Softwareinformationssystems. Dazu wird zunächst ein Überblick über die Zielstellung des Softwareinformationssystems sowie die Möglichkeiten seiner Umsetzung gegeben. Des Weiteren werden die grundlegenden Forderungen an das Design der Architektur und die wesentlichen Aufgaben der einzelnen Komponenten vorgestellt.

3.4.1 Motivation

Mit der rasanten Entwicklung und den neuen Möglichkeiten in der Informations- und Kommunikationstechnologie ergeben sich neue Herausforderungen bei der Projektierung und Dimensionierung von Industrieanlagen. Die Auswirkungen sind in allen Stufen der Anlagentechnik, der Feldbus- und Steuerungsebene bis hin zur Leitebene zu spüren. So werden einheitliche Infrastruktur- und Kommunikationssysteme, Steuerungs- und Automatisierungssysteme mit deutlich gesteigerter Rechenleistung, intelligente, selbstdiagnostizierbare Sensoren und Aktuatoren sowie neue Methoden zur Regelung- und Überwachung von Prozessen integriert. Mit ausgewählten Worten kann die Vision der weiteren Technisierung und Automatisierung folgendermaßen beschrieben werden.

„Die Produktionsanlage der Zukunft soll sich selbstständig überwachen, Verschleiß fördernde Betriebszustände, Leistungseinschränkungen sowie Instandhaltungsbedarf automatisch erkennen und diese so frühzeitig melden, dass die Nutzung der Anlage nicht beeinträchtigt wird.“[1]

Um die Verwirklichung dieses Zieles voranzutreiben, sollen mit dieser Arbeit Grundlagen für die Entwicklung eines modularen Systems von Data-Mining-Werkzeugen zur Generierung von entscheidungsfähigem Wissen für Überwachungs- und Planungsaufgaben mit dem Einsatzschwerpunkt an Gurtförderanlagen, aber auch für andere Bereiche der Prozessindustrie geschaffen werden. Diese Lösung soll es ermöglichen, die Standardüberwachungs- und -diagnosemodule, die bisher auf verschiedenen Ebenen, wie der

- Prozessebene (PCS), in die Echtzeitdatenerfassung, Steuerung, Regelung und Monitoring integriert werden
- Produktionsebene (MES), in der im Wesentlichen Produktions- und Prozessoptimierung und -überwachung, Qualitätskontrolle und -überwachung stattfinden, und

- Betriebsebene (ERP), mit der Betriebs- und Produktionsplanung, Wartungs- und Instandhaltungsplanung und Kostenoptimierung,

verteilt sind und oft separat bestehen, durch die neuen im Rahmen dieses Projekts erarbeiteten Verfahren, in ein innovatives Anlagenmanagementsystem zu integrieren. Aus den ermittelten funktionalen und nichtfunktionalen Anforderungen sowie den Richtlinien und Konventionen, welche für das Softwaredesign bzw. für die Softwarearchitektur gelten, soll der Vorschlag für die zukünftige Softwarearchitektur entstehen.

3.4.2 Aufgabenstellung und Ziele

Das zu entwickelnde Softwaresystem soll Prozessdaten mithilfe datenbasierter Methoden automatisch untersuchen und Fehler detektieren. Die Auswertung erfolgt in der Regel zyklisch mit stets neu eintreffenden, aktualisierten Daten. Die dafür benötigten Überwachungsparameter (Modelle und Grenzwerte) erlernt das System selbstständig mit dem zu entwickelnden Teach-Modul bzw. sie werden manuell konfiguriert. Die Prozessdaten können aus den verschiedensten Quellen stammen, wie z. B. aus einem datenbankbasierten Prozessdatenarchiv, aus CSV-Dateien oder von OPC-Server (OPC - OLE for Process Control, OLE - Object Linking and Embedding) Schnittstellen. Die Auswertung der Prozessdaten kann unabhängig von der im Hintergrund laufenden zeitlichen Steuerung auch manuell herbeigeführt werden. Generell wird es jeweils ein Graphisches-User-Interface (GUI) sowohl für das Modul zum Anlernen (Teach) als auch für die Auswertung und Anzeige der Ergebnisse (Runtime) geben. Ebenfalls gefordert ist eine Überwachungsoption bzw. „is alive“ (Lebenszeichen) Funktion. Die Software ist so auszulegen, dass

- sie intuitiv bedienbar ist, d. h. die Nutzer lediglich Grundkenntnisse im Umgang mit Software benötigen
- die zwei wesentlichen Aufgaben realisierbar sind. das Anlernen des Normalverhaltens sowie die Überwachung mit Fehlerdetektion, -lokalisierung und -identifizierung
- weitere Methoden mit vertretbarem Aufwand im Customizing ergänzbar sind.

Das Teach-Modul

In dem Programmteil für die Konfiguration und das Training werden durch eingewiesene/geschulte Mitarbeiter die Grundeinstellungen für die Software bezüglich der Anlagenspezifika vorgenommen. Im Trainingsmode erfasst das Modul automatisch den Normalzustand einer Anlage und berechnet selbstständig die notwendigen statistischen Modellparameter sowie Grenz-/Schwellwerte für die Überwachung. Aus

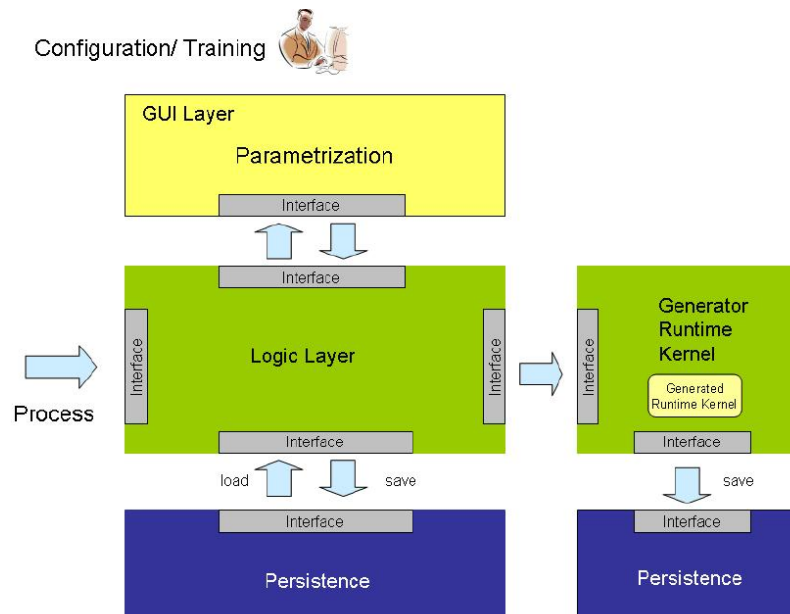


Abb. 3.8: Prinzipieller Aufbau des Teach-Moduls

Das Runtime-Modul

Das Runtime-Modul läuft üblicherweise auf einem Rechner in der Nähe der Anlage. Es berechnet aus den einlaufenden Prozessdaten statistische Kennwerte und vergleicht diese mit dem statistischen Modell und den Kennwerten des Normalzustandes (Runtime-Kernel). Bei Abweichungen wird eine Fehlermeldung erzeugt, anschließend beginnt die Fehleridentifizierung. Bekannte Fehler werden automatisch mit ihrem Namen und möglichen Handlungsanweisungen für das Service-Personal ausgegeben. Noch nicht bekannte Fehler sind als neu zu kennzeichnen. Das Modul lernt automatisch die zugehörigen Fehlermuster. Das Personal vergibt für den neuen Fehler einen Namen. Danach wird das System diesen Fehler in Zukunft automatisch identifizieren. Das Runtime-Modul kann wie in Abbildung 3.9 dargestellt aufgebaut sein.

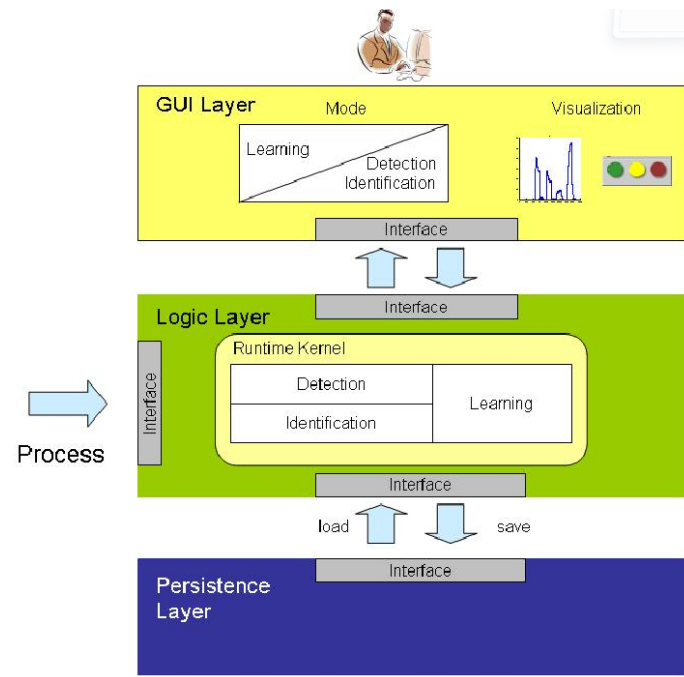


Abb. 3.9: Prinzipieller Aufbau des Runtime-Moduls

Mit den Möglichkeiten des Anlernens und Überwachens ist der in Abbildung 3.10 dargestellte Ablauf zum automatisierten Wissensaufbau und zur Überwachung denkbar.

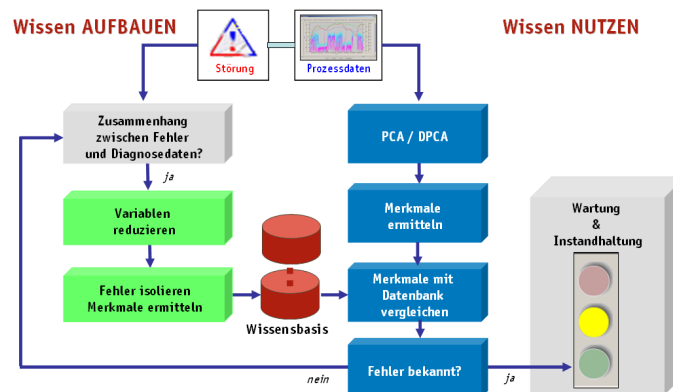


Abb. 3.10: Ablauf - Überwachung und Wissensaufbau

Weitere Anforderungen und Rahmenbedingungen

Das System soll sowohl auf einer autarken Board-Unit als auch auf einem separaten Server installierbar und lauffähig sein. Die sich daraus ergebenden differenzierten Anforderungen an die Architektur und an das Design der Software sind zu berücksichtigen.

Für die mathematische Implementierung der Analysemethoden ist das Einbinden von Fremdbibliotheken oder Programmen vorzusehen, da grundlegende mathematische Algorithmen nicht neu programmiert werden sollen. Des Weiteren ist der Aspekt der Austauschbarkeit von bestimmten Softwareteilen zu berücksichtigen.

Das System soll so konzipiert sein, dass es branchenspezifische Aufgaben komplett abdecken kann oder aber über definierte Schnittstellen mit anderen Systemen (z. B. mit BDE-, PLM-, oder ERP-Systemen) zusammenarbeitet und somit eine flexible Anpassung an unterschiedliche Anforderungen möglich wird. Darüber hinaus ist die Systemstruktur offen zu gestalten.

Bei der Implementierung der Software ist manuelle Adaption an die einzelnen Anlagen erforderlich, deren Umfang vor allem von den Engineering-Erfahrungen des Anlagenpersonals abhängig ist. Die Prozessdatenerfassung muss leistungsfähig ausgelegt sein. Die Daten werden direkt von Sensoren und Aktuatoren geliefert und per Prozessdatenschnittstelle (OPC - OLE for Process Control, Datenbank, Datei) übernommen.

3.4.3 Anforderungen an das Design der Softwarearchitektur

3.4.3.1 Architektursichten

In der heutigen Zeit hat sich das Konzept verschiedener Architektursichten durchgesetzt. Wurden in den 70er Jahren noch zwei Sichten gefordert, die Baustein- und Laufzeitsicht, gelten heute laut Kruchten [41] und Starke [66] drei bis fünf Sichten als angemessen. In der Praxis haben sich zusätzlich die Kontextsicht und die Verteilungssicht bewährt.

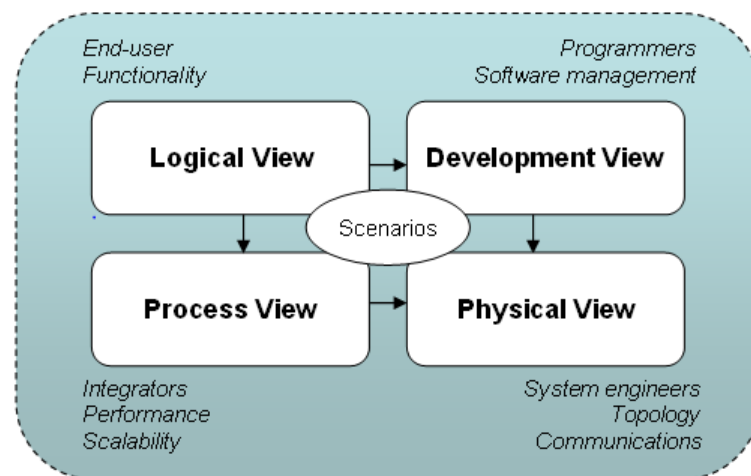


Abb. 3.11: Das 4+1 Sichten-Modell nach P. Kruchten [41]

Die Beschreibung der Sichten ist eine wichtige Voraussetzung für die folgenden Schritte des Designkonzeptes und der Implementierung der Software. Starke und Hruschka entwarfen in [67] das arc42-Template zur Dokumentation von Software- und IT-Architekturen als Grundlage für den iterativen Entstehungsprozess einer Softwarearchitektur. Die darin enthaltenen Schritte und Richtlinien bilden auch die Grundlage für den im Rahmen dieser Arbeit entworfenen Architekturentwurf. Ein wichtiges Design-Hilfsmittel ist die Modellierungssprache Unified Modeling Language (UML), entwickelt unter anderem von der ISO (ISO/IEC 19501 für Version 2.1.2) und der Object Management Group (OMG), welche geeignete Diagramme und Dokumentationen dieser Sichten zur Verfügung stellt.

Neben den bereits aufgeführten Zielen und Randbedingungen sind folgende Aspekte zu klären und zu begründen:

- die Kontextabgrenzung
- die verschiedenen Sichten
- übergreifende Architekturaspekte und technische Konzepte sowie
- Entwurfsentscheidungen.

Die komplexeste Aufgabe während der Entwicklung einer neuen Software ist erfahrungsgemäß die Übertragung der Anforderungsspezifikation in eine Softwarearchitektur. In den letzten Jahren konnte man beobachten, dass eine genaue Spezifikation, Analyse und ein gutes Design existenziell wichtig für den Entwicklungs- und auch für den späteren Lebenszyklusprozess einer Software sind. Sowohl im Entwicklungs- als auch im evolutionären Prozess ist das Designen der Software ein immer wiederkehrender bedeutender Arbeitsschritt. Die nachfolgende Abbildung 3.12 verdeutlicht die wesentlichen Etappen grafisch.

Entscheidend für den Erfolg des Designentwurfes sind die spezifizierten funktionalen und Nicht-funktionalen Anforderungen, welche die Software erfüllen soll. Diese sind in einer Requirements-Spezifikation zu dokumentieren.

3.4.3.2 Modularität und Aufteilung in Schichten

Eine grundlegende Forderung an die Software ist, dass sie modular aufgebaut sowie horizontal und vertikal gut strukturiert sein soll. Damit kann die Komplexität der Abhängigkeiten und die Kopplung zwischen den Modulen und Schichten reduziert werden. Deshalb wird ein objektorientierter Entwurf sowie die Wahl einer Multi-Tier Architektur empfohlen. Den prinzipiellen Aufbau einer Multi-Tier Architektur stellt die Abbildung 3.13 dar.

Die konkrete Anzahl und der Inhalt der Schichten hängen unter anderem von den Aufgaben sowie den Einsatz- und Umgebungsbedingungen der Software ab. Hierbei unterscheidet man unter anderem zwischen Präsentationsschicht, Steuerungsschicht, Anwendungsschicht, Datenzugriffsschicht oder auch Persistenzschicht. Inhalt einer

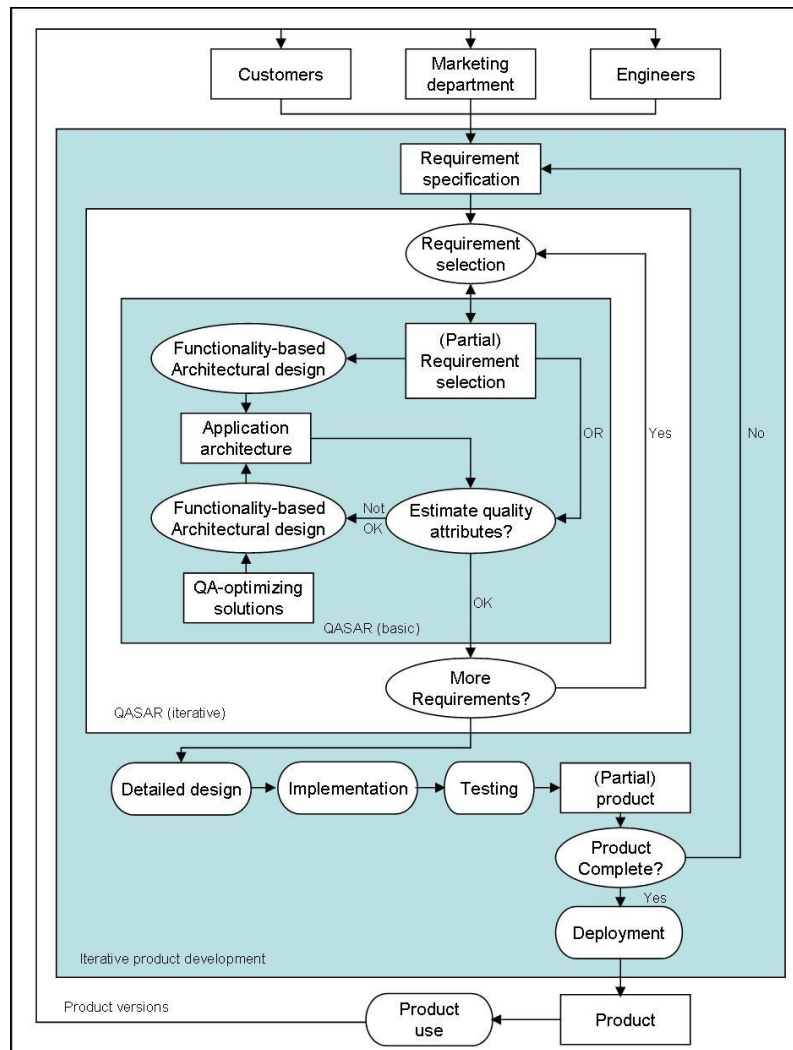


Abb. 3.12: Evolutionärer Prozess der Softwarearchitektur laut Bosch [4]

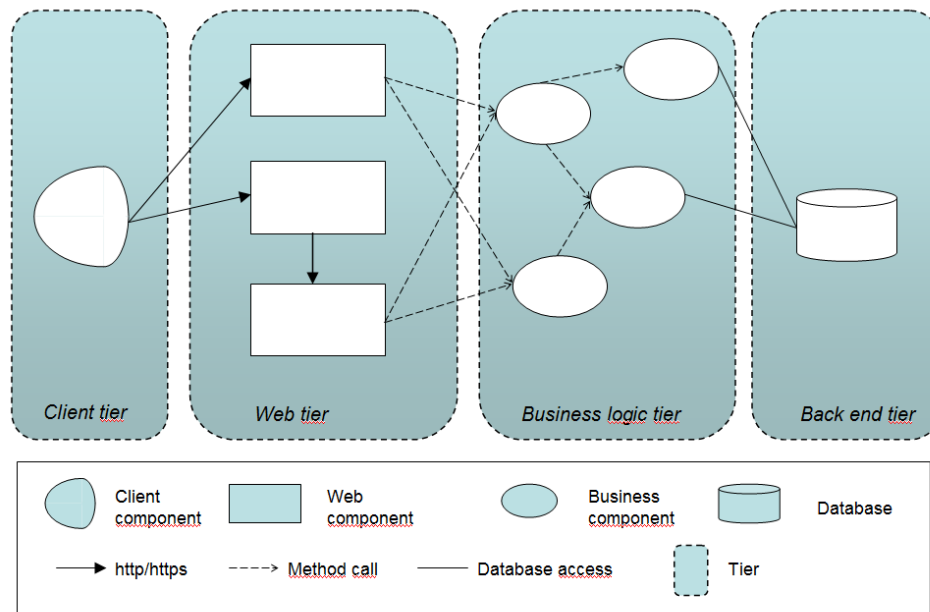


Abb. 3.13: Aufbau einer Multi-Tier Architektur laut [9]

Präsentationsschicht kann beispielsweise das im nachfolgenden Abschnitt „Nutzung von Patterns und Frameworks“ beschriebene Model-View-Controller Entwurfsmuster sein.

Module werden in objektorientierten Systemen vor allem durch Klassen gebildet. Beim Entwurf eines Systems sind im Normalfall zuerst die fachlichen Gegenstände oder Belange zu Klassen zu abstrahieren. Betrachtet man nur diese fachliche Dimension, so ist jedes auf diese Weise entstandene Modul für genau einen Belang zuständig. Allerdings müssen in einem realen System auch die technischen und nicht-funktionalen Belange berücksichtigt werden. Diese können teilweise in Klassen gekapselt werden, jedoch oft handelt es sich um Querschnittsfunktionen, die zu den fachlichen Klassen hinzugefügt werden müssen. Sie verteilen sich also über die fachliche Klassenstruktur und sind untrennbar mit den fachlichen Belangen vermischt.

3.4.3.3 Nutzung von Patterns und Frameworks

Mit der Idee, der Verwendung von frei nutzbaren Frameworks und Design Patterns, soll vor allen Dingen eine Zeit und Kostenersparnis bei der Entwicklung der Software erzielt sowie die Fehleranfälligkeit reduziert werden.

Design Patterns

Als Design Patterns oder auch Entwurfsmuster werden Lösungsmuster von Softwaredesignern bezeichnet, denen aufgefallen ist, dass man viele Probleme auf ähnliche

Weise lösen kann. Auf diese Weise entstanden eine Vielzahl von Regelwerken mit Lösungsmustern, die eine optimale Wiederverwendung von Bausteinen ermöglichen. Design-Patterns ziehen sich durch die gesamte Java-Klassenbibliothek. Zu den bekanntesten gehören das Beobachter- (Observer-) Pattern, Singleton, Fassade, Fabrik (Factory) und Composite. Diese und viele weitere wurden unter anderem durch M. Fowler in [23] beschrieben. Ein dem Observer-Muster sehr ähnliches Pattern ist das etwas erweiterte, unter dem Namen MVC (Model-View-Controller) bekannte Pattern, welches ein ganz wesentliches Konzept zur Programmierung grafischer Benutzeroberflächen ist. Es dient zur Entkopplung von Fachobjekten und deren Darstellung und wird hier beispielhaft erläutert. Die nachfolgende Abbildung 3.14 stellt den prinzipiellen Aufbau des MVC-Pattern dar.

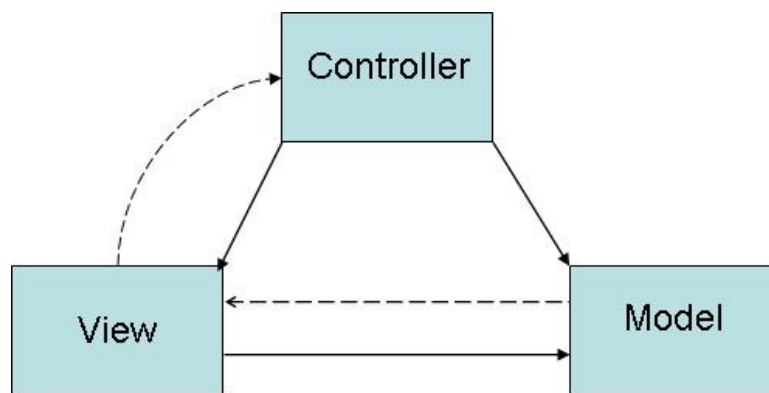


Abb. 3.14: MVC Pattern

Das MVC-Pattern besteht aus dem Model, der View und dem Controller, deren grundlegende Aufgaben folgende sind:

Model

- Es kapselt fachliche Objekte der Anwendung.
- Das Model kennt seine Views und Controller, die sich an- und abmelden können.
- Es benachrichtigt angemeldete Komponenten bei Zustandsänderungen.

View

- Sie dient der Darstellung und Präsentation (meistens visuelles GUI, inklusive Bedienelemente. mehrere Views sind gleichzeitig möglich).
- Die View bereitet Informationen vom Model für den Benutzer auf und stellt diese geeignet dar.
- Sie erzeugt und initialisiert seinen Controller.
- Sie realisiert einen Update-Callback.

Controller

- Optional kann zu jeder View jeweils ein Controller die Entgegennahme und Koordination von Benutzerinteraktionen steuern.
- Er akzeptiert Benutzereingaben als Nachrichten.
- Der Controller übersetzt Nachrichten in Anfragen an Model oder View.
- Er realisiert einen Update-Callback.

Die üblichen Methoden des MVC-Patterns sind `show()`, `updateModel()`, `updateView()`.

Frameworks

Die Effektivität der Softwareentwicklung kann durch den Einsatz von Frameworks deutlich verbessert werden. Sie bieten den Vorteil, dass sie bei Design und Implementierung bereits fertig entworfene und implementierte Softwarestrukturen mitbringen. Ungeeignete Entwurfsentscheidungen werden so von vornherein vermieden. Folgende weitere Vorteile sind u. a. damit verbunden:

- Die wesentlichen Entwurfsprobleme sind schon gelöst.
- Der Code ist leichter zu verstehen, vor allem wenn fremde Entwickler damit umgehen müssen
- Frameworks erleichtern die Arbeit im Team.
- In der Regel existiert bereits eine ausführliche Dokumentation für das Framework.
- Der mit einem Framework erstellte Code ist effizient und im Großen und Ganzen fehlerfrei.
- Frameworks bieten den derzeit höchsten Grad an Wiederverwendung mit einem Maximum an Produktivität und Robustheit.

Deshalb ist beim Entwurf und bei der Implementierung die Verwendung existierender Frameworks zu berücksichtigen.

3.4.3.4 Konventionen der Softwareentwicklung

Während des Entstehungsprozesses einer Software sind durch die Entwickler eine Reihe von Konventionen einzuhalten. Dazu gehören auch die Programmierrichtlinien entsprechend den internen Code- und Style-Guidelines. Diese sind dem Entwickler in Form von Eclipse-Plug-ins (findbugs, pmd, checkstyle) zwingend vorgeschrieben. Die Einhaltung der Regeln wird durch die Plug-ins während des Entwickelns und in den nächtlichen Build-Läufen automatisch überprüft. Die Vielzahl der Regeln und deren Bedeutung sind im Detail zu finden unter: Bug Descriptions [68], Regeln zur Code-Erstellung [10], Einhaltung von Codierungsrichtlinien in Java Programmen [64].

Einige der wichtigsten Aspekte, deren Einhaltung durch die Plug-ins gewährleistet wird, sind:

- Beachtung von Sichtbarkeiten
- keine Überdeckung von lokalen Variablen
- Vermeidung von public Data-Member
- Deklaration lokaler Variablen erst dann, wenn sie gebraucht werden
- Regeln für die Benennung von Methoden und Klassen
- einheitliche Formatierung
- Verwendung von generischen Datentypen zur Erhöhung der Typensicherheit
- Vermeidung von Ausgaben auf die Konsole (Nutzung von Logging-Frameworks).

Des Weiteren haben übergreifende Architekturaspekte Einfluss auf den Architektur-entwurf, wie beispielsweise die Persistenz, die Wahl der Benutzungsoberfläche, die Ablaufsteuerung, die Transaktionsbehandlung, die Session-Behandlung viele weitere, nachzulesen in [67].

3.4.4 Zusammenfassung

Die Grundlagen für den Erfolg bei der Entwicklung eines neuen Softwaresystems werden bei der Aufnahme der Anforderungen und Umgebungsbedingungen, beim Design der Architektur sowie bei der Festlegung verwendeter Bibliotheken und Frameworks gelegt. Für eine Vielzahl von softwaretechnischen Problemstellungen gibt es bereits Lösungen bzw. Entwürfe von Lösungen. Die Nutzung dieser Lösungsentwürfe in Form von Design-Patterns sowie von vorhandenen Frameworks senkt die Fehleranfälligkeit, macht die Software robust gegenüber Änderungen und steigert die Produktivität bei der Entwicklung. Die Aufteilung der Software in Schichten und die Modularisierung hilft ebenfalls die Fehleranfälligkeit der Software zu senken und erleichtert den Austausch bzw. die Erweiterung von Teilen der Software. Diese Fakten und Vorteile haben Einfluss auf den in Kapitel 5 vorgestellten Entwurf des Softwareinformationssystems.

3.5 Die Gurtförderanlagen als komplexes mechatronisches System

3.5.1 Grundlegender Aufbau

Schüttgut-Materialflusssysteme sind verkettete, häufig sehr unterschiedliche Fördermaschinen und -anlagen für die Aufnahme, den Transport, die Lagerung und den

Umschlag von Schüttgütern. Unter den Bedingungen des Bergbaus sind die verfahrenbaren und stationären Maschinenkomplexe hohen Beanspruchungen ausgesetzt, müssen in einem 24h-Betrieb verfügbar sein und sollen eine hohe Lebensdauer aufweisen. Stillstandszeiten bedeuten in allen Fällen nicht nur Kosten für die Instandhaltung, sondern vielfach höhere Kosten für den Produktionsausfall bzw. nicht hinnehmbare Versorgungsengpässe. Das stellt höchste Anforderungen sowohl an den Maschinen- und Anlagenbauer als auch an den Service-Dienstleister, der innerhalb des Portfolios auf neue Dienstleistungskonzepte bezüglich Fehlerfrüherkennung und Störungsbeseitigung zugreifen muss.

Die Gurtförderanlage als industrielles Massentransportsystem realisiert den Transport von Schüttgütern innerhalb eines verketteten technischen Systems, muss also als logistisches Teilsystem angesehen werden. Da sie in der Regel nicht redundant ausgelegt ist, stellt sie ein Single Point of Failure dar, denn ihre Verfügbarkeit hat eine enorm hohe Bedeutung für die Funktion des gesamten Systems, für den Gesamtprozess. Dementsprechend werden zur Sicherstellung einer hohen Verfügbarkeit in der Industrie große Anstrengungen auf dem Gebiet Fehlerdetektion und Fehlerbeseitigung unternommen.

Eine Gurtförderanlage besteht im Wesentlichen aus folgenden Baugruppen:

- Fördergurt
- Antriebsstation
- Umlenkstation
- Tragrollen
- Fördergut,

wobei Tragrollen und Fördergurt als Bestandteil der Bandstraße angesehen werden können. Nachfolgende Abbildung 3.15 ist eine schematische Darstellung des Aufbaus einer Gurtförderanlage. Der Fördergurt verbindet die Antriebs- mit der Umlenkstation. Er läuft von der Umlenkstation zur Antriebsstation im sogenannten Obertrum und von der Antriebsstation zur Umlenkstation im Untertrum auf den Tragrollen.

- Die hier aufgelisteten Einflussgrößen sind nur Beispiele und nicht vollständig. Weitergehende Informationen dazu sind in [35] zu finden.

In [35] wurde ein analytisches Modell einer Gurtförderanlage beschrieben, welches integraler Bestandteil eines modellgestützten Fehlerdiagnosesystems für Gurtförderanlagen ist. Im Ergebnis dieser Arbeit entstand das Modell einer Gurtförderanlage mit diskreten Massen und den angreifenden Kräften. In der nachfolgenden Abbildung 3.16 ist der prinzipielle Aufbau des Modells dargestellt.

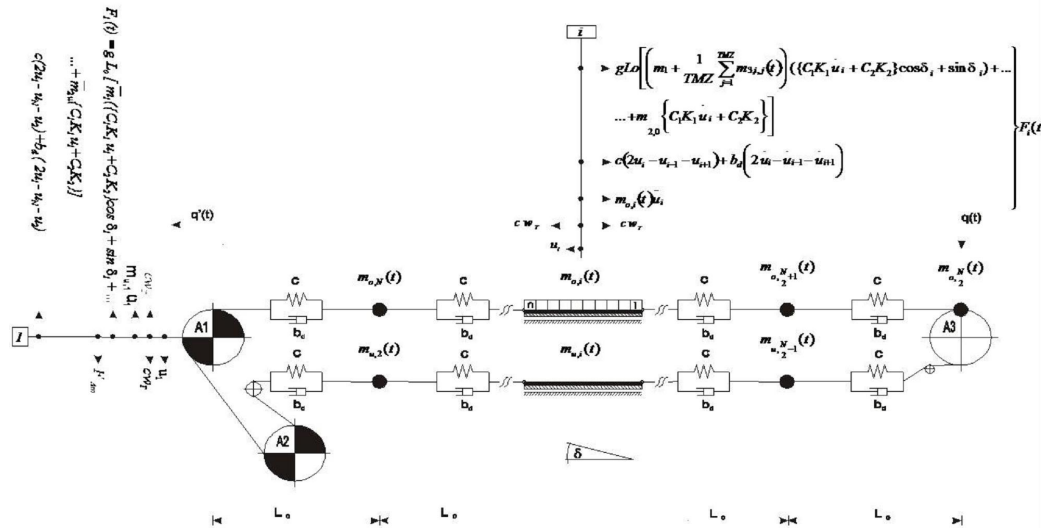


Abb. 3.16: Modell einer Gurtförderanlage mit diskreten Massen aus [35]

Für die Fehlererkennung auf Basis multivariater Verfahren ist ein analytisches Modell, wie in [35] beschrieben nicht notwendig. Es ist aber wichtig die gemessenen Ein- und Ausgangsgrößen, wesentliche Einflussgrößen auf das Prozessverhalten sowie den grundlegenden Prozessverlauf zu kennen.

3.5.2 Der Prozess

Prinzipiell lassen sich industrielle Prozesse nach ihrem Verhalten folgendermaßen unterscheiden:

- stationäres Verhalten - charakterisiert durch keine oder wenig Dynamik in den Größen
- dynamisches Verhalten, mit verhältnismäßig viel Dynamik in den Größen während des Überganges zwischen verschiedenen Arbeitspunkten.

Stationärer Prozess bedeutet, dass sich die Wahrscheinlichkeitsverteilungen und Abhängigkeiten der Messgrößen im Zeitverlauf nicht ändern dürfen. Der Prozess ist über lange Zeiträume gesehen im Prinzip ein nicht-stationäres Signal. Die Veränderungen vollziehen sich jedoch i.d.R. langsam, sodass ein quasi-stationäres Signal entsteht. Das bedeutet, dass bestimmte Abschnitte als stationär angesehen werden können.

Damit erhält man die Möglichkeit, Verfahren der Zeitreihenanalyse, welche stationäres Verhalten voraussetzen, für bestimmte Abschnitte zu nutzen. Die Zeitpunkte der Arbeitspunktwechsel bei dynamischem Verhalten können zeit- oder ereignisbestimmt sein.

Bei einer Gurtförderanlage verursachen An- und Abfahrvorgänge der Anlage, Be- und Entladung des Gurtes bzw. starke Änderungen der Massen auf dem Gurt, hohe Dynamik bzw. Wechsel in Arbeitspunkten. Die Abbildung 3.17 stellt das Verhalten der Prozessmessgrößen Leistung und Vibration in Abhängigkeit der Ereignisse Ab- und Anfahren sowie Beladung dar. Dabei können folgende Bereiche unterschieden werden:

- zwischen Sample-Nr. 0 bis 270 - stationärer Zustand
- ab Sample 270 - Beginn des Anhaltevorganges
- ab Sample 340 - Anfahrvorgang mit Beladung der Bandanlage
- ab Sample 370 - stationärer Prozess.

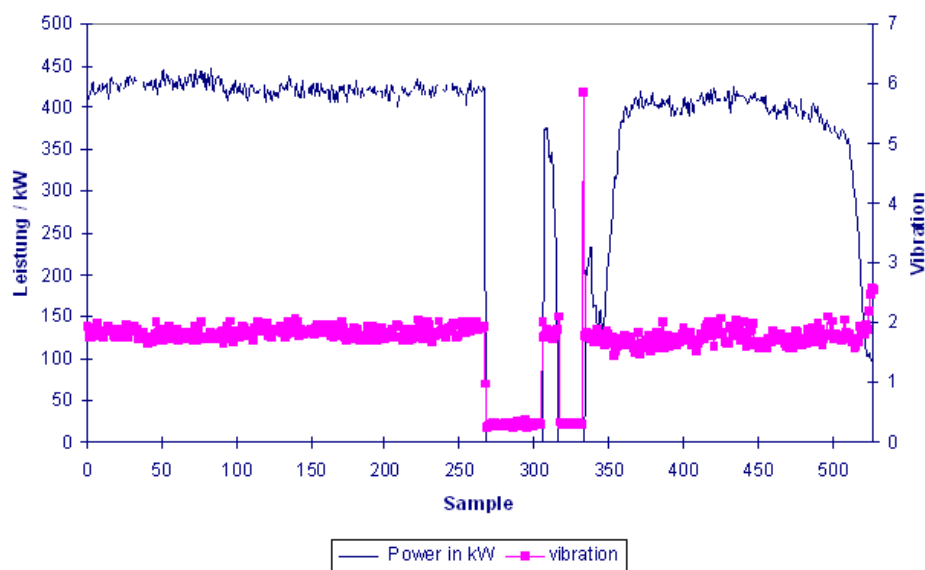


Abb. 3.17: Dynamischer Prozess mit ereignisbestimmten Arbeitspunktwechseln (Bsp. Gurtförderanlage)

3.5.3 Fehlerzustände

Mithilfe der statistisch basierten Methoden zur Fehlerdetektion sollen typische aber auch außergewöhnliche Fehler sowie Störungen, die Einfluss auf den nominalen Betrieb der Gurtförderanlage haben bzw. kritische Betriebszustände hervorrufen, erkannt werden und das ohne eine spezielle analytische Beschreibung von bestimmten Fehlerzuständen.

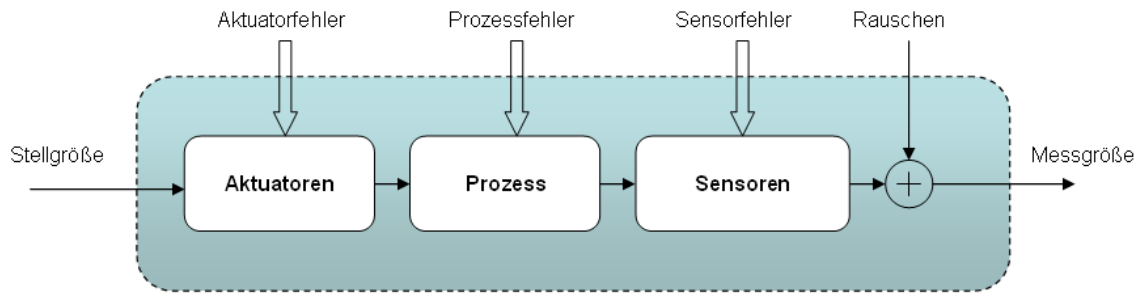


Abb. 3.18: Schematische Darstellung eines Prozesses unter Berücksichtigung von Fehlern

Zu den typischen Fehlern und Störungen gehören u.a.:

- Schieflläufe
- Anbackungen
- Überschüttungen
- Sensorfehler.

Außergewöhnliche Fehler sind beispielsweise:

- der Ausfall einzelner Komponenten, wie z.B. Ausfall oder Teilfunktion des Antriebssystems, deren Ursachen im Getriebe, im Motor, im Lager oder in der Antriebstrommel selbst liegen kann
- die schlechte Justierung der Anlage oder Bauteilen, z.B. infolge eines Shiftings (Rücken) der Anlage oder nach dem Austausch einzelner Komponenten, wie etwa nach dem Wechsel einer Trommel.

3.6 Zusammenfassung

Eine Gurtförderanlage kann man als typisches mechatronisches System bezeichnen. Alle im Abschnitt 2.3 benannten wesentlichen Bestandteile, wie mechanische und elektromechanische Komponenten, Sensoren, Aktoren aber auch das System zu Informationsverarbeitung (Steuerung und Regelung) sind wiederzufinden. Der Prozess mit seinen dynamischen und stationären Bereichen ist typisch für die Prozessindustrie, aber nicht vergleichbar mit verfahrenstechnischen Prozessen. Insofern ist er als Benchmark-Prozess für den Bereich der Prozessindustrie geeignet.

4 Entwicklung datenbasierter Verfahren zur Anlagenüberwachung

Die in dem letzten Kapitel vorgestellten datenbasierten Methoden werden bezüglich ihrer Anwendbarkeit an Prozessdaten und Fehlerzuständen einer Gurtförderanlage untersucht. Nach der Erläuterung der bei der Anwendung der „Standard-Methoden“ aufgetretenen Probleme, werden sinnvolle Anpassungen und Erweiterungen der Algorithmen vorgestellt. Diese sollen trotz der Schwierigkeiten mit den realen Messdaten und dem Prozessverhalten eine optimale Fehlerdetektion an Gurtförderanlagen ermöglichen.

4.1 Problemformulierung

Bei der Anwendung von datenbasierten Methoden der Fehlerdetektion an realen industriellen Prozessen stellt man oftmals fest, dass sich die Daten und Prozessbedingungen deutlich von den bei der Entwicklung der Methoden getroffenen Annahmen zur Prozessdynamik und Verteilung der Daten unterscheiden. Die Effizienz und Zuverlässigkeit der Methoden leiden u.a. aus zwei wichtigen Gründen:

- Die Mittelwerte der Prozessgrößen verändern sich im Laufe der Zeit, zeigen dynamisches Verhalten
- eine Reihe von Messwerten haben einen ähnlichen Informationsgehalt, das heißt, sie sind in bestimmten Teilen redundant.

Diese Probleme zeigten sich auch bei der Anwendung der in Kapitel 4 vorgestellten datenbasierten Methoden zur Fehlerdetektion an den Prozessdaten einer Gurtförderanlage. Im Rahmen dieser Arbeit werden nun verschiedene Algorithmen zur Datenaufbereitung untersucht, welche eine effektive Arbeit der datenbasierten Methoden zur Fehlerdetektion ermöglichen.

4.2 Datenaufbereitung

4.2.1 Verminderung von Dynamiken im Erwartungswert durch lineare Kombination von Messgrößen

Die Untersuchungen von Messdaten industrieller Anlagen haben gezeigt, dass unter bestimmten Bedingungen Dynamiken im Erwartungswert als Störgrößen vorhanden sind. Die Ursachen dafür liegen beispielsweise in dem Einbau von Sensoren in verschiedenen Komponenten, die aber konstruktiv miteinander verbunden sind. So können bei einer Antriebseinheit einer Gurtförderanlage zwei Motoren an einer Welle befestigt sein. Im fehlerfreien Betrieb (Normalbetrieb) erzeugen die Stromsensoren beider Motoren Daten mit einem ähnlichen Verlauf. Weichen die Stromdaten plötzlich voneinander ab, so liegt mit hoher Wahrscheinlichkeit ein Fehlerfall vor. Des Weiteren liefern, soweit vorhanden, die Leistungs-, Momenten- und Drehzahlsensoren beider Motoren nahezu identische Datenverläufe. Selbst zwischen Strom, Leistung und Moment ist ein starker Zusammenhang zu bemerken.

Die Redundanz zweier Variablen kann so gedeutet werden, dass sie durch einen nicht beobachtbaren Faktor (latenter Faktor) bewirkt wird, der hinter diesen Variablen steht. Die Messdaten der Variablen weisen hohe Korrelationen zu diesen Faktoren selbst auf. Korrelieren viele Messdaten miteinander, bringt das in Bezug zur Fehlererkennung keinen Informationsgewinn. Im Ergebnis der PCA entstehen Eigenvektoren mit Eigenwerten, deren darauf projizierte Daten keine Korrelation aufweisen. Dabei werden aber die sogenannten Hauptkomponenten (Eigenvektoren mit dem größten Varianzanteil) durch die korrelierenden deterministischen Anteile dominiert, während die anderen, zufälligen oder auch Fehleranteile eher einen verschwindend kleinen Einfluss haben.

Die beiden nachfolgenden Diagramme verdeutlichen die vorhandenen deterministischen Anteile in den Messdaten industrieller Prozesse.

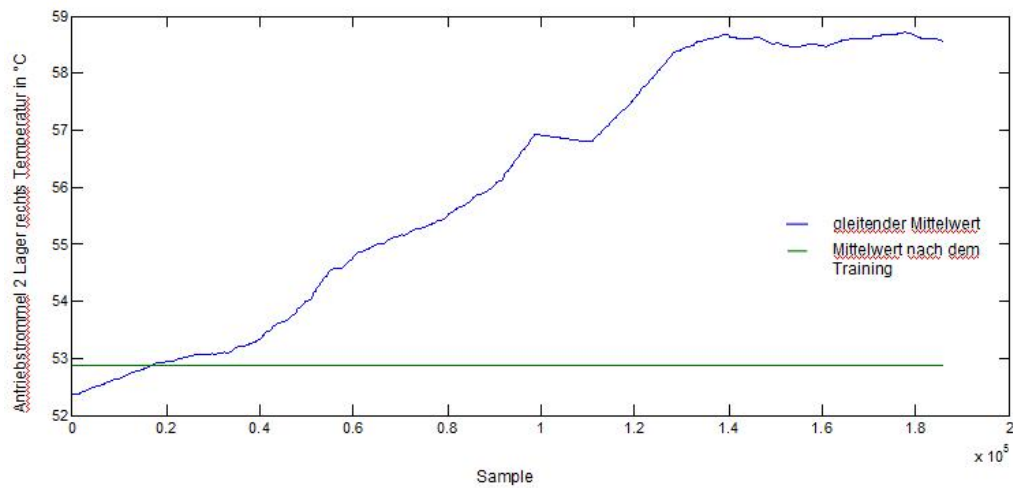


Abb. 4.1: Zeitverlauf des Mittelwertes der Lagertemperatur Antriebsstrommel 2 der Bandantriebsstation gegenüber dem Mittelwert des angelernten fehlerfreien Zustandes von Dezember bis Februar

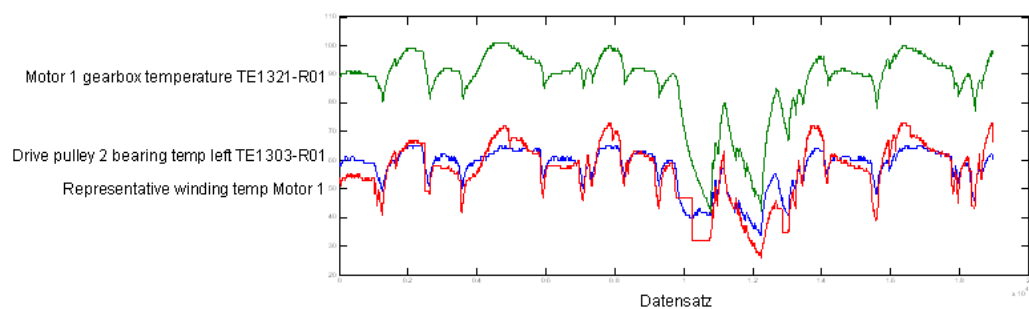


Abb. 4.2: Zeitverlauf ausgewählter Signale einer Bandantriebsstation

Durch die standardmäßig durchgeführte Normierung der Signale lassen sich diese Einflüsse nicht eliminieren. Beim Einsatz von z.B. Fehlerdetektionsverfahren auf Basis der PCA können bestehende dynamische Vorgänge im Erwartungswert die Detektionsperformanz stark beeinträchtigen. Lösen lassen sich diese Probleme einerseits durch zusätzlich eingebaute Hardwareredundanzen oder andererseits durch eine weitere Vorverarbeitung der Messwerte, z.B. gebildete Linearkombinationen der Signale. Nachfolgend sei zunächst das Prinzip der Hardwareredundanzen und danach das der Linearkombination von Messwerten beschrieben.

a) Hardwareredundanz

Zeitreihen kann man prinzipiell in einen periodischen Anteil (z.B. Tag-Nacht-Rhythmus), auch als Trend (langfristige Entwicklung) bezeichnet und einen auf zufällige Einflüsse rückführbaren Anteil, den sogenannten Rest oder irregulären Anteil oder auch Fehler, zerlegen. Die Beschreibung einer Zeitreihe sieht folgendermaßen aus:

$$x_k = E(x(k)) + \varepsilon_x, \quad (4.1)$$

mit $E(x(k))$ dem periodischen Anteil und ε_x dem Fehleranteil.

Gegeben sind das Sensorsignal $x = E(x(k)) + \varepsilon_x$ und das redundante Sensorsignal mit $\bar{x} = E(\bar{x}(k)) + \varepsilon_{\bar{x}}$. Für den Erwartungswert gilt

$$E(x(k)) = E(\bar{x}(k)) \quad (4.2)$$

folglich:

$$\Delta x = x - \bar{x} = \varepsilon_x - \varepsilon_{\bar{x}} \quad (4.3)$$

$$E(\Delta x(k)) = 0 \quad (4.4)$$

Hardwareredundanzen sind kostspielig, aufwendig und machen nur in wenigen Ausnahmefällen Sinn.

b) Linearkombination von Messwerten

Die folgenden Zeilen beschreiben die Idee, dass über eine lineare Kombination von Messgrößen, z.B. über die Differenzbildung, die Dynamiken im Erwartungswert ebenfalls minimiert bzw. zu null gesetzt werden können (siehe Abbildung 4.3).

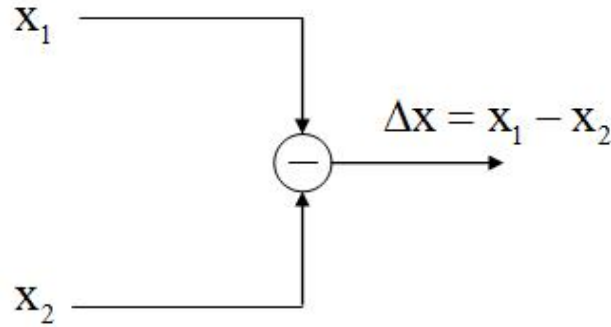


Abb. 4.3: Prinzip der Differenzsignalbildung

Eine lineare Kombination von Messgrößen, wie die Differenzbildung, kann durch $\alpha^T x \in \mathcal{R}$ mit $\alpha \in \mathcal{R}^m$ und $x \in \mathcal{R}^{m \times 1}$ beschrieben werden. Für $\alpha^T x$ ergibt sich

$$\alpha^T x = \alpha^T (E(x(k)) + \varepsilon_x) = \alpha^T \varepsilon_x, \quad (4.5)$$

wenn $\alpha^T E(x(k)) = 0$ ist. $\alpha^T E(x(k))$ beschreibt/modelliert hier die Korrelation.

$$\alpha^T E(x(k)) = 0 \quad (4.6)$$

Nach der Differenzbildung bleibt der zufällige Anteil erhalten. Voraussetzung ist aber, dass die unterschiedlichen Signale bezüglich Mittelwert = 0 und Standardabweichung = 1 normiert werden, was im Allgemeinen bei der PCA und GLR der Fall ist. Im Ergebnis erhält man eine mittelwertfreie, stochastische Größe.

Zur Entscheidungsfindung, zwischen welchen Signalen eine Differenzbildung sinnvoll ist, kann man die Korrelation zwischen den Signalen ermitteln. Mithilfe der Korrelationsanalyse bestimmt man den Zusammenhang von Variablenpaaren in einer Datenmenge. Der Korrelationskoeffizient bietet eine Aussage über die Stärke der Streuung der Variable um eine lineare Verknüpfung. Er kann Werte zwischen -1 und 1 annehmen, wobei bei einem Wert nahe 0 kein linearer Zusammenhang zwischen den Werten besteht. Der Korrelationskoeffizient zweier Variablen x_1 und x_2 berechnet sich über die Formel

$$r_{x_1, x_2} = \frac{\text{cov}(x_1, x_2)}{\text{std}(x_1) \text{std}(x_2)}, \quad (4.7)$$

wobei $\text{cov}(x_1, x_2)$ die Kovarianz zwischen den Variablen x_1 und x_2 ist. Ein Korrelationskoeffizient von 0,9 und mehr sagt aus, dass zwischen den betrachteten Variablen mit hoher Wahrscheinlichkeit ein linearer Zusammenhang besteht. Aber er sagt nicht aus, um wie viel x_1 bei jeder Erhöhung von x_2 anwächst bzw. sinkt. Des Weiteren kann man über einen großen Korrelationskoeffizienten nicht automatisch auf die Ursache für den Zusammenhang schließen, also ob die Ursache für das Sinken oder Steigen von x_1 bei x_2 liegt oder umgekehrt. Bei der Bildung einer Kovarianzma-

trix mit normierten Zeitreihen, wie Abschnitt PCA 3.3.1 beschrieben, entspricht die Kovarianzmatrix der Korrelationsmatrix.

Die nachfolgende Beispiel-Matrix enthält die Korrelationskoeffizienten von Messsignalen einer Bandantriebsstation. Bei allen Werten mit einem Korrelationskoeffizienten größer 0,9 kann das Differenzsignal für die Diagnose herangezogen werden. Bei mehreren korrelierenden Signalen sollte aber mindestens ein Originalsignal erhalten bleiben.

Drive pulley 2 bearing T left TE1303-R01	1	0,7	0,7	0,46	0,46	0,46	0,46	0,57	0,94	0,72	0,73	0,91	0,72	0,7	0,71
Drive pulley 2 representative vibration	0,7	1	0,99	0,38	0,38	0,37	0,37	0,52	0,57	0,97	0,45	0,62	0,97	0,97	0,97
Drive pulley 2 vibration left VE1403-B01	0,7	0,99	1	0,41	0,41	0,41	0,4	0,55	0,57	0,96	0,46	0,63	0,96	0,97	0,97
Motor 1 Average Phase Current in A	0,46	0,38	0,41	1	1	1	1	0,89	0,34	0,47	0,31	0,43	0,47	0,44	0,42
Motor 1 Phase A Current in A	0,46	0,38	0,41	1	1	0,99	1	0,89	0,34	0,46	0,31	0,42	0,46	0,43	0,42
Motor 1 Phase B Current in A	0,46	0,37	0,41	1	0,99	1	0,99	0,88	0,35	0,46	0,31	0,43	0,46	0,43	0,41
Motor 1 Phase C Current in A	0,46	0,37	0,4	1	1	0,99	1	0,88	0,34	0,46	0,31	0,42	0,46	0,43	0,41
Motor 1 Real Power in kW	0,57	0,52	0,55	0,89	0,89	0,88	0,88	1	0,39	0,6	0,4	0,52	0,6	0,57	0,55
Motor 1 gearbox temperature TE1321-R01	0,94	0,57	0,57	0,34	0,34	0,35	0,34	0,39	1	0,58	0,72	0,88	0,58	0,56	0,58
Vibration gearbox 1 Sensor 2 VE1422-B01	0,72	0,97	0,96	0,47	0,46	0,46	0,46	0,6	0,58	1	0,43	0,64	1	0,98	0,97
Representative bearing T Motor 1	0,73	0,45	0,46	0,31	0,31	0,31	0,31	0,4	0,72	0,43	1	0,88	0,43	0,45	0,47
Representative winding T Motor 1	0,91	0,62	0,63	0,43	0,42	0,43	0,42	0,52	0,88	0,64	0,88	1	0,64	0,63	0,64
Vibration gearbox 1 Sensor 1 VE1421-B01	0,72	0,97	0,96	0,47	0,46	0,46	0,46	0,6	0,58	1	0,43	0,64	1	0,98	0,97
Vibration motor 1 VE1411-B01	0,7	0,97	0,97	0,44	0,43	0,43	0,43	0,57	0,56	0,98	0,45	0,63	0,98	1	0,98
Vibration motor 1 VE1430-B01	0,71	0,97	0,97	0,42	0,42	0,41	0,41	0,55	0,58	0,97	0,47	0,64	0,97	0,98	1

Abb. 4.4: Korrelationsmatrix für die Messsignale einer Bandantriebsstation

Um die Auswirkung der Differenzbildung zwischen korrelierenden Signalen bei der PCA zu verdeutlichen, wird nachfolgend die Differenzbildung bei einer Datenmatrix mit zwei Variablen untersucht. Die Daten sind bezüglich ihres Mittelwertes und Standardabweichung normalisiert.

$$Z = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \\ \vdots & \vdots \\ z_{N1} & z_{N2} \end{bmatrix} \in \mathcal{R}^{N \times 2}. \quad (4.8)$$

Über den Zusammenhang 3.13 erhält man die Kovarianzmatrix.

$$\Sigma = \begin{bmatrix} z_1^T z_1 & z_2^T z_1 \\ z_1^T z_2 & z_2^T z_2 \end{bmatrix} \quad (4.9)$$

Die Hauptdiagonalen enthalten die Varianzen von z_1 und z_2 und die Nebendiagonalen die Kovarianzen, wobei $z_1^T z_2 = z_2^T z_1$ ist. Zur Bestimmung der Eigenwerte sind die Gleichungen

$$\det(\Sigma - I\Lambda) = 0 \quad (4.10)$$

und

$$\det \begin{bmatrix} (z_1^T z_1 - \Lambda) & z_2^T z_1 \\ z_1^T z_2 & (z_2^T z_2 - \Lambda) \end{bmatrix} = 0 \quad (4.11)$$

zu lösen.

Folglich erhält man

$$\lambda_{1,2} = \frac{z_1^T z_1 + z_2^T z_2}{2} \pm \frac{1}{2} \sqrt{(z_1^T z_1)^2 + 2z_1^T z_1 z_2^T z_2 + (z_2^T z_2)^2 - 4z_1^T z_1 z_2^T z_2 - 4(z_1^T z_2)^2} \quad (4.12)$$

und nach der Zusammenfassung

$$\lambda_{1,2} = \frac{z_1^T z_1 + z_2^T z_2}{2} \pm \frac{1}{2} \sqrt{(z_1^T z_1 - z_2^T z_2)^2 - 4(z_1^T z_2)^2}. \quad (4.13)$$

Wie in 4.1 dargestellt setzen sich unsere Signale aus einem periodischen Anteil und aus einem Fehleranteil zusammen. Das bedeutet für z_1 und z_2 :

$$z_1 = m + \varepsilon_1 \quad (4.14)$$

$$z_2 = n + \varepsilon_2. \quad (4.15)$$

Des Weiteren kann davon ausgegangen werden, dass $m \gg \varepsilon_1$ und $n \gg \varepsilon_2$.

Um die Auswirkung der Differenzbildung bei stark korrelierenden Signalen auf die Eigenwerte zu untersuchen, werden nachfolgend drei verschiedene Fälle betrachtet:

a) keine Differenzbildung zwischen z_1 und z_2

Dadurch, dass z_1 und z_2 stark korrelieren, ist $z_1^T z_1 \approx z_2^T z_2$. Das heißt, man kann $z_2^T z_2$ durch $z_1^T z_1$ ersetzen. Für die Eigenwerte ergibt sich:

$$\lambda_{1,2} \approx z_1^T z_1 \pm z_1^T z_2. \quad (4.16)$$

Letztendlich heißt das, dass sich die Eigenwerte durch den doppelten Kovarianzanteil unterscheiden. Da die Kovarianzen bei den stark korrelierenden Signalen ebenfalls relativ groß sind, erhält man zwei sich von den Beträgen stark unterscheidende Eigenwerte.

b) Ersetzen von z_2 durch \tilde{z} , der Differenz aus z_1 und z_2 ohne nachfolgende Normierung

Infolge der Differenzbildung heben sich die deterministischen Anteile auf und man erhält $\tilde{z} = \varepsilon_1 - \varepsilon_2$, wobei $z_1 \gg \tilde{z}$ ist. Setzt man nun \tilde{z} in 4.13 ein, so erhält man:

$$\lambda_{1,2} \approx \frac{1}{2} z_1^T z_1 \pm \frac{1}{2} z_1^T \tilde{z}. \quad (4.17)$$

Folglich ist $\lambda_1 \approx z_1^T z_1$ also etwa die Varianz von z_1 und $\lambda_2 \approx 0$. Beide Eigenwerte unterscheiden sich erneut stark von den Beträgen, wobei der große Eigenwert von dem deterministischen Anteil bestimmt wird.

c) Ersetzen von z_2 durch \tilde{z} , aber mit nachfolgender Normierung von \tilde{z}

Durch die Normierung von \tilde{z} auf Mittelwert null und Standardabweichung eins erreicht \tilde{z} eine ähnliche Größenordnung wie z_1 , das heißt $\tilde{z} \approx z_1$. Nach dem Einsetzen in 4.13 erhält man

$$\lambda_{1,2} = \frac{z_1^T z_1 + \tilde{z}^T \tilde{z}}{2} \pm \frac{1}{2} \sqrt{(z_1^T z_1 - \tilde{z}^T \tilde{z})^2 - 4(z_1^T \tilde{z})^2}, \quad (4.18)$$

mit

$$\tilde{z}^T \tilde{z} = \varepsilon_1^T \varepsilon_1 - 2\varepsilon_1^T \varepsilon_2 + \varepsilon_2^T \varepsilon_2. \quad (4.19)$$

Gleichung 4.19 in 4.18 eingesetzt ergibt

$$\begin{aligned} \lambda_{1,2} = & \frac{1}{2} (z_1^T z_1 + \varepsilon_1^T \varepsilon_1 - 2\varepsilon_1^T \varepsilon_2 + \varepsilon_2^T \varepsilon_2) \pm \\ & \pm \frac{1}{2} \sqrt{(z_1^T z_1 - \varepsilon_1^T \varepsilon_1 + 2\varepsilon_1^T \varepsilon_2 - \varepsilon_2^T \varepsilon_2)^2 - 4(z_1^T (\varepsilon_1 - \varepsilon_2))^2}. \end{aligned} \quad (4.20)$$

Da $\varepsilon_1^T \varepsilon_2$ und $z_1^T (\varepsilon_1 - \varepsilon_2)$ nahe null sind, kann man Gleichung 4.20 folgendermaßen vereinfachen:

$$\lambda_{1,2} = \frac{1}{2} (z_1^T z_1 + \varepsilon_1^T \varepsilon_1 + \varepsilon_2^T \varepsilon_2) \pm \frac{1}{2} (z_1^T z_1 - \varepsilon_1^T \varepsilon_1 - \varepsilon_2^T \varepsilon_2). \quad (4.21)$$

So ergeben sich für $\lambda_1 = z_1^T z_1$ und $\lambda_2 = \varepsilon_1^T \varepsilon_1 + \varepsilon_2^T \varepsilon_2$. Die Eigenwerte der beiden Vektoren kann man nun als in etwa gleichwertig betrachten. Während der eine durch den deterministischen Anteil bestimmt wird, sind bei dem anderen die Varianzen der Rauschanteile die entscheidenden Faktoren.

Nach Betrachtung der drei Fälle a) bis c) kann man schlussfolgern, dass durch die Differenzbildung stark korrelierender Signale und nachfolgende Normierung des entstandenen Differenzsignals eine bessere Detektierbarkeit von Fehlern, die nicht im deterministischen Anteil liegen, gegeben ist.

Im Folgenden wurde nun der Einfluss der Differenzsignalbildung auf die PCA und die im Anschluss an die PCA gebildeten Indizes T^2 und SPE für multiplikative und additive Fehlerbilder untersucht. Von den drei erzeugten Datensätzen x_1 , x_2 und x_3 , korrelieren x_1 und x_2 miteinander, während x_3 unkorreliert zu diesen ist. Die additiven und multiplikativen Fehler wurden mit einer Stärke von 10% aufgeprägt, was 3σ entspricht. Laut [58] befindet sich ein Prozess noch „In-control“, solange die Abweichung nicht größer als 3σ ist. Der fehlerhafte Zustand ist in diesen Beispielen immer ab Sample 251 in den Daten vorhanden.

Zur Verdeutlichung der Wirksamkeit der PCA mit und ohne Differenzsignalbildung erfolgte das Aufprägen der additiven und multiplikativen Fehler in drei verschiedenen Varianten.

1. Additiver und multiplikativer Fehler nur auf Signal x_1

Im Diagramm sehen die Daten folgendermaßen aus:

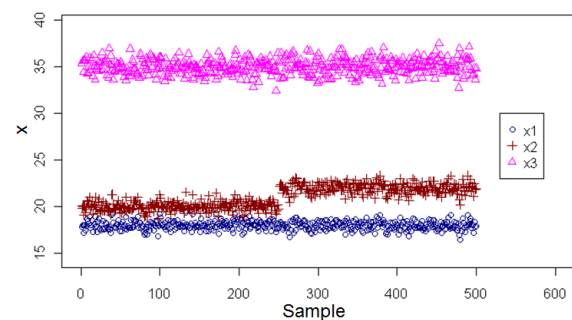
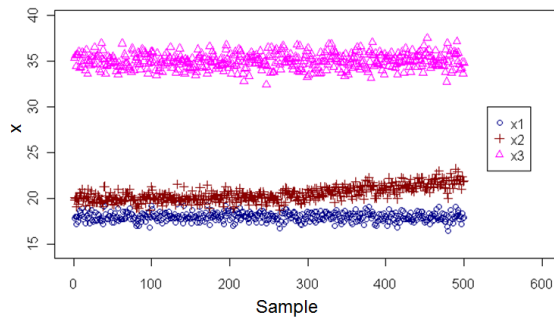


Abb. 4.5: Daten mit aufgeprägtem multiplikativen Fehler

Abb. 4.6: Daten mit aufgeprägtem additivem Fehler

Anschließend wurden die Daten von x_1 , x_2 und x_3 auf den Mittelwert = 0 und der Standardabweichung = 1 normiert.

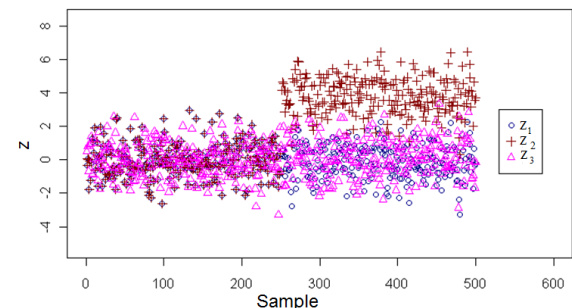
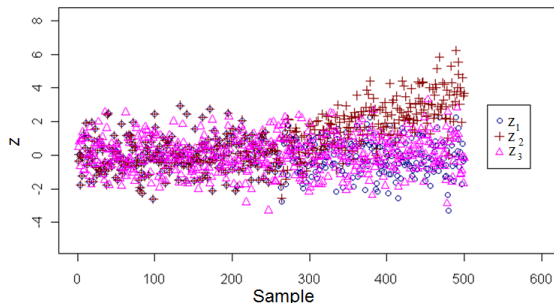


Abb. 4.7: Daten mit multiplikativen Fehler nach der Normierung von Mittelwert und Standardabweichung

Abb. 4.8: Daten mit additivem Fehler nach der Normierung von Mittelwert und Standardabweichung

Danach werden wieder die drei oben beschriebenen Fälle betrachtet.

a) PCA ohne die Bildung von Differenzsignalen

Wie im Fall 1 beschrieben, wird zunächst die PCA an den unveränderten Daten (keine Differenzbildung) durchgeführt. Die Eigenwerte und Indizes sehen folgendermaßen aus:

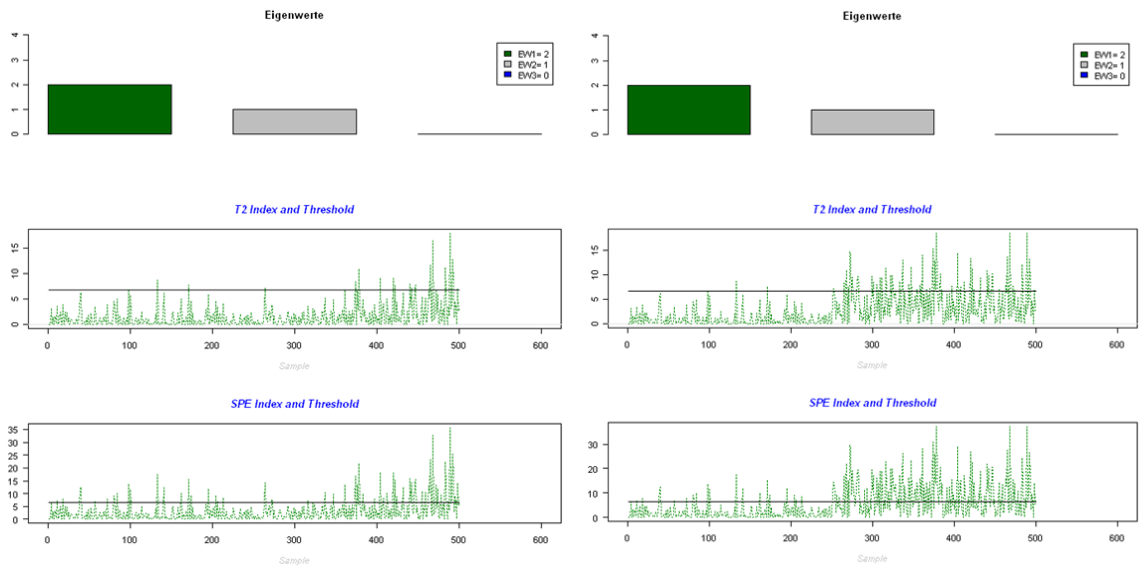


Abb. 4.9: Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler ohne Differenzsignalbildung

Abb. 4.10: Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler ohne Differenzsignalbildung

Im Ergebnis der PCA hat man die im Fall a) beschriebene starke Unterscheidung der Eigenwerte. Durch die starke Dominanz der deterministischen Anteile werden die Grenzwerte zwar überschritten, aber trotz des vorhandenen Fehlers kehrt der Index immer wieder in den Normalbereich zurück. Insbesondere der multiplikative Fehler wird nicht eindeutig erkannt.

- b)** PCA mit Differenzsignal $\tilde{z} = z_1 - z_2$ ohne nachfolgende Normierung von \tilde{z}
 Die nachfolgenden Diagramme enthalten die normierten Werte für z_1 und z_3 sowie die nicht normierten Werte von \tilde{z} .

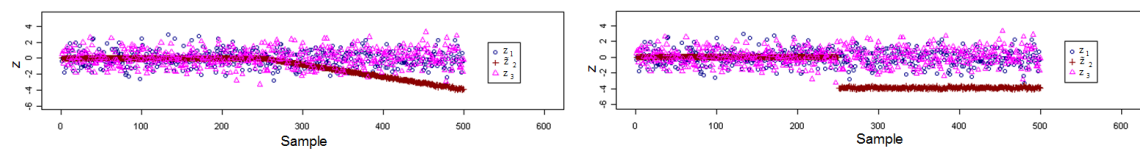


Abb. 4.11: Daten mit multiplikativem Fehler und nicht normiertem Differenzsignal

Abb. 4.12: Daten mit additivem Fehler und nicht normiertem Differenzsignal

Die Eigenwerte und Indizes sehen folgendermaßen aus:

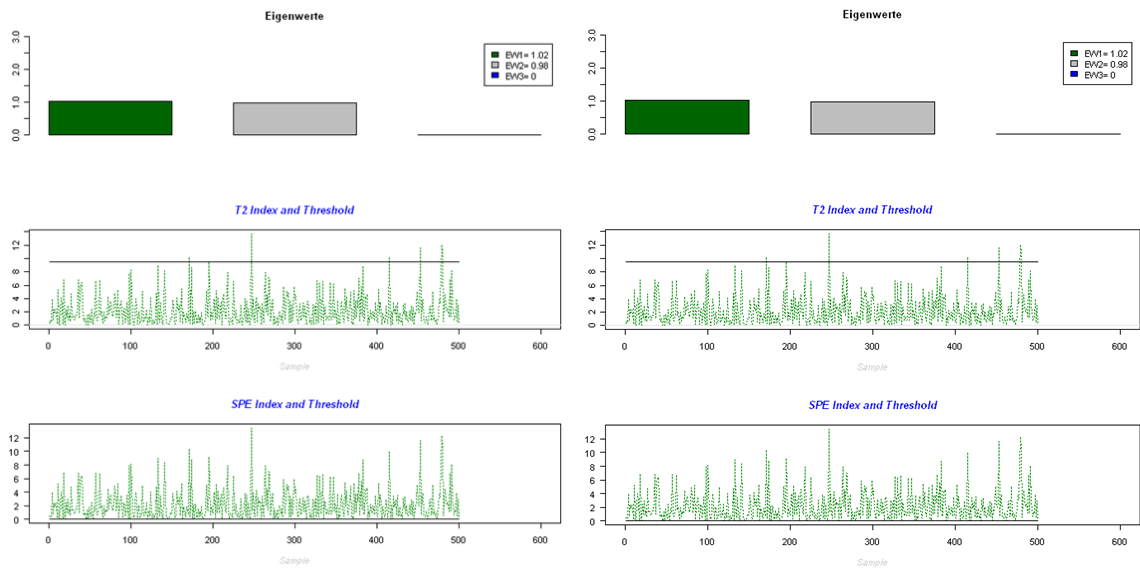


Abb. 4.13: Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler mit nicht normiertem Differenzsignal

Abb. 4.14: Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler mit nicht normiertem Differenzsignal

In diesem Fall werden weder der multiplikative noch der additive Fehler erkannt. Der aufgeprägte Fehleranteil geht bei der Dominanz der anderen deterministischen Anteile unter.

- c) PCA mit Differenzsignal $\tilde{z} = z_1 - z_2$ mit nachfolgender Normierung von \tilde{z}
Die nachfolgenden Diagramme enthalten die normierten Werte für z_1 und z_3 sowie die nachträglich normierten Werte von \tilde{z} .

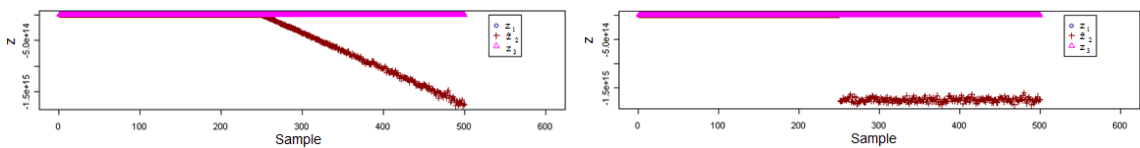


Abb. 4.15: Daten mit multiplikativem Fehler und normiertem Differenzsignal

Abb. 4.16: Daten mit additivem Fehler und normiertem Differenzsignal

Die Eigenwerte und Indizes sehen folgendermaßen aus:

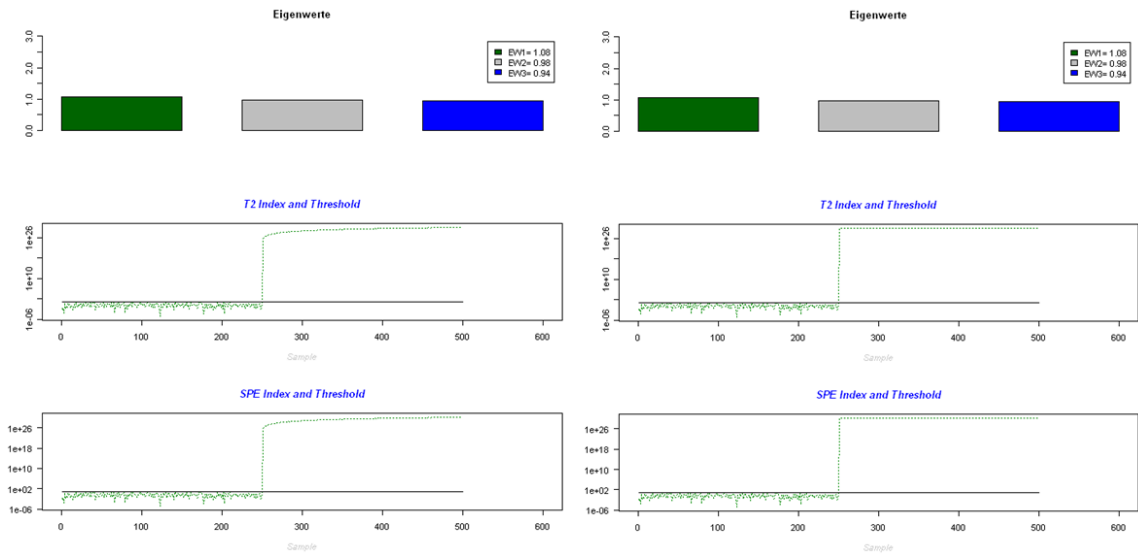


Abb. 4.17: Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler mit normiertem Differenzsignal

Abb. 4.18: Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler mit normiertem Differenzsignal

Die Differenzbildung und nachfolgende Normierung des Fehleranteils führt, wie im Fall c) beschrieben zu einer stärkeren Bewertung, was sich in den etwa gleichgroßen Varianzanteilen bei allen drei Eigenvektoren bemerkbar macht. Sowohl der multiplikative als auch der additive Fehler werden nun eindeutig erkannt.

2. Additiver und multiplikativer Fehler auf Signal x_1 und x_2 - Fehlersignale von x_1 und x_2 unterscheiden sich nicht

Bei dieser Variante wurde bei beiden korrelierenden Signalen der Sensoren x_1 und x_2 zum Rauschen der gleiche Fehler aufgeprägt. Die Daten haben nun den folgenden Verlauf:

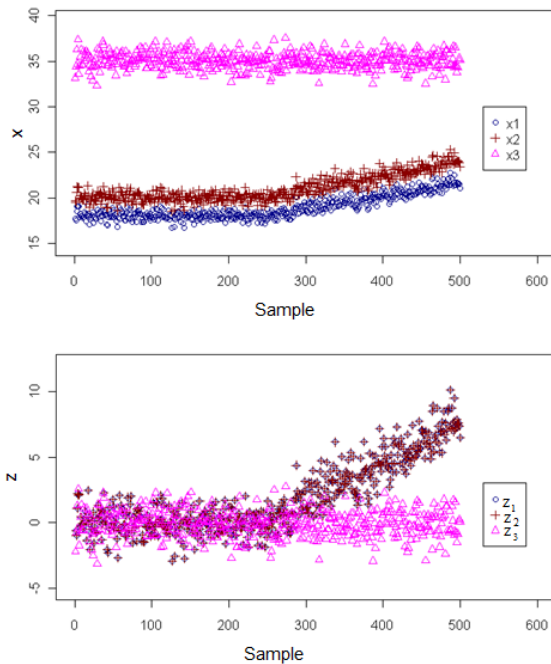


Abb. 4.19: Daten mit aufgeprägtem multiplikativem Fehler auf x_1 und x_2 vor und nach der Normierung

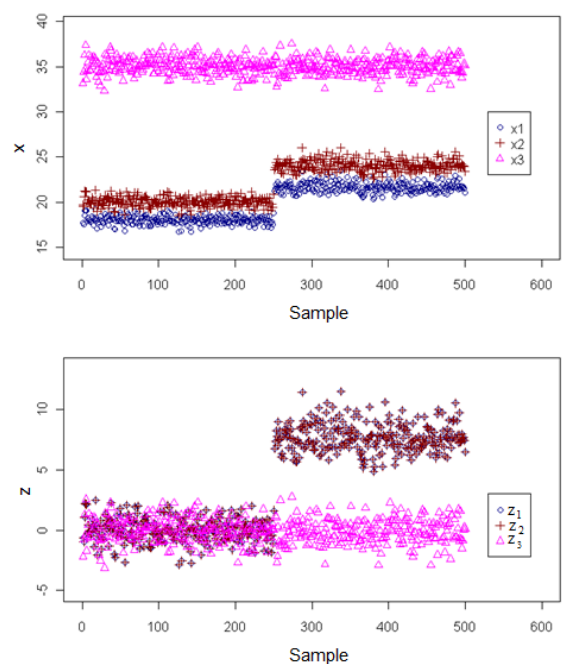


Abb. 4.20: Daten mit aufgeprägtem additivem Fehler auf x_1 und x_2 vor und nach der Normierung

Es wurde nur der oben beschriebene Fall c) betrachtet, da bereits unter 1 gezeigt werden konnte, dass bei den Fällen a) und b) die Fehler entweder schlecht oder gar nicht erkannt werden.

- c) PCA mit Differenzsignal $\tilde{z} = z_1 - z_2$ mit nachfolgender Normierung von \tilde{z}
Die nachfolgenden Diagramme enthalten die normierten Werte für z_1 und z_3 sowie die nachträglich normierten Werte von \tilde{z} .

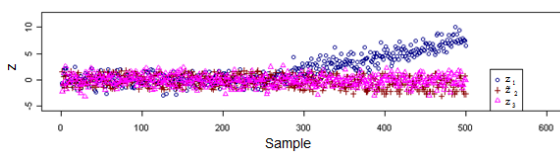


Abb. 4.21: Daten mit multiplikativem Fehler und normiertem Differenzsignal

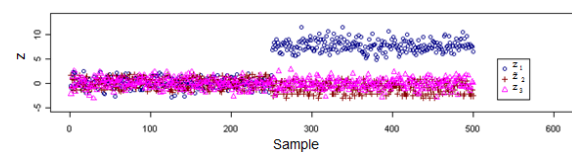


Abb. 4.22: Daten mit additivem Fehler und normiertem Differenzsignal

Die Eigenwerte und Indizes sehen folgendermaßen aus:

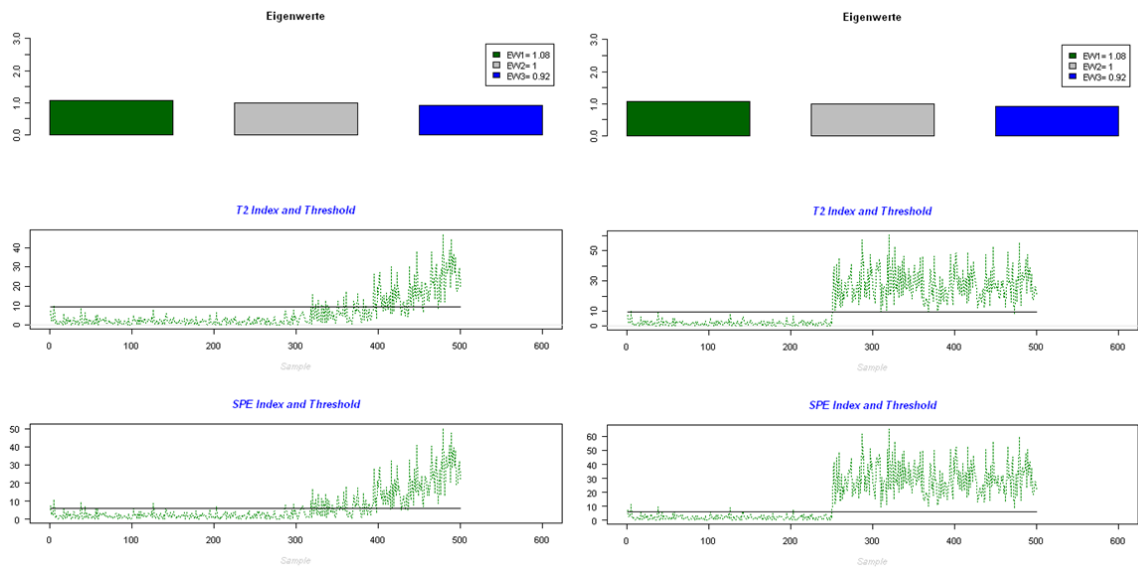


Abb. 4.23: Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler mit normiertem Differenzsignal

Abb. 4.24: Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler mit normiertem Differenzsignal

Deutlich wird, dass trotz der Differenzbildung über die sich sowohl im deterministischen als auch im Rauchanteil gleich verhaltenden Signale x_1 und x_2 beide Fehler erkannt wurden. Beim additiven Fehler überschritt der Index den Grenzwert deutlich und eher als beim multiplikativen Fehler.

3. **Additiver und multiplikativer Fehler auf Signal x_1 und x_2 - Fehler-signale von x_1 und x_2 unterscheiden sich** Abschließend wurden beiden korrelierenden Signalen der Sensoren x_1 und x_2 zum Rauschen unterschiedliche Fehler aufgeprägt.

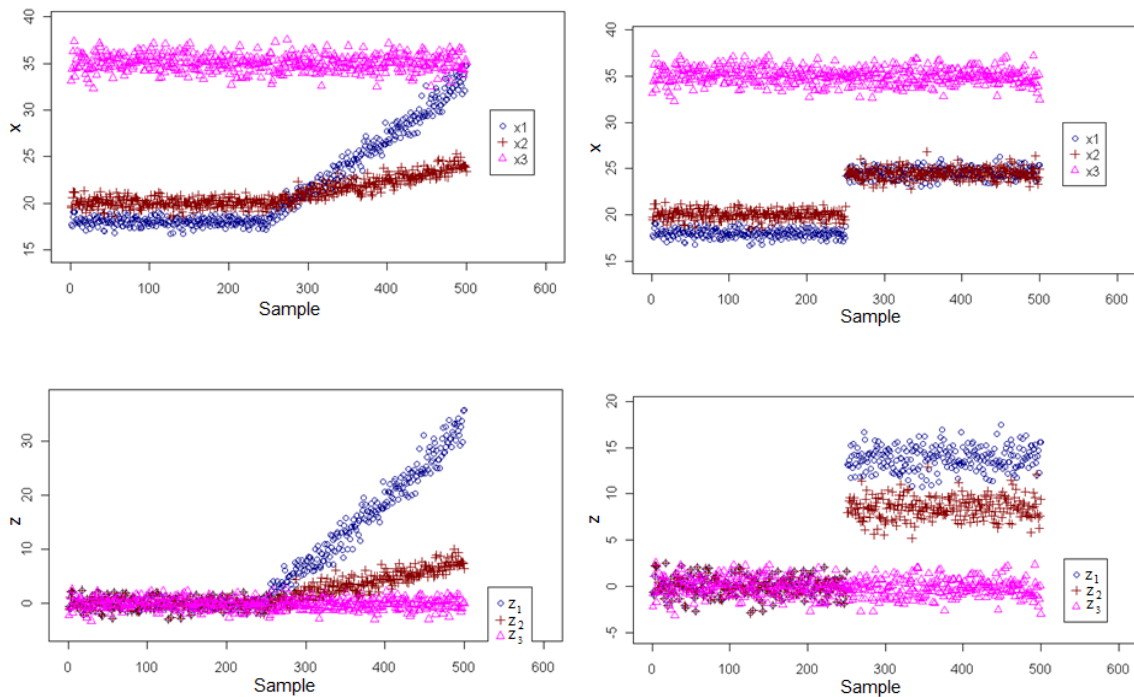


Abb. 4.25: Daten mit aufgeprägtem multiplikativem Fehler auf x_1 und x_2 vor und nach der Normierung

Abb. 4.26: Daten mit aufgeprägtem additivem Fehler auf x_1 und x_2 vor und nach der Normierung

Für diese Variante wurde ebenfalls nur der oben beschriebene Fall c) betrachtet.

- c)** PCA mit Differenzsignal $\tilde{z} = z_1 - z_2$ mit nachfolgender Normierung von \tilde{z}
 Die nachfolgenden Diagramme enthalten die normierten Werte für z_1 und z_3 sowie die nachträglich normierten Werte von \tilde{z} .

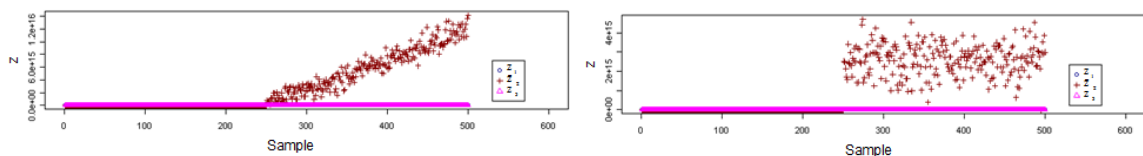


Abb. 4.27: Daten mit multiplikativem, nicht korrelierendem Fehler und normiertem Differenzsignal

Abb. 4.28: Daten mit additivem, nicht korrelierendem Fehler und normiertem Differenzsignal

Die Eigenwerte und Indizes sehen folgendermaßen aus:

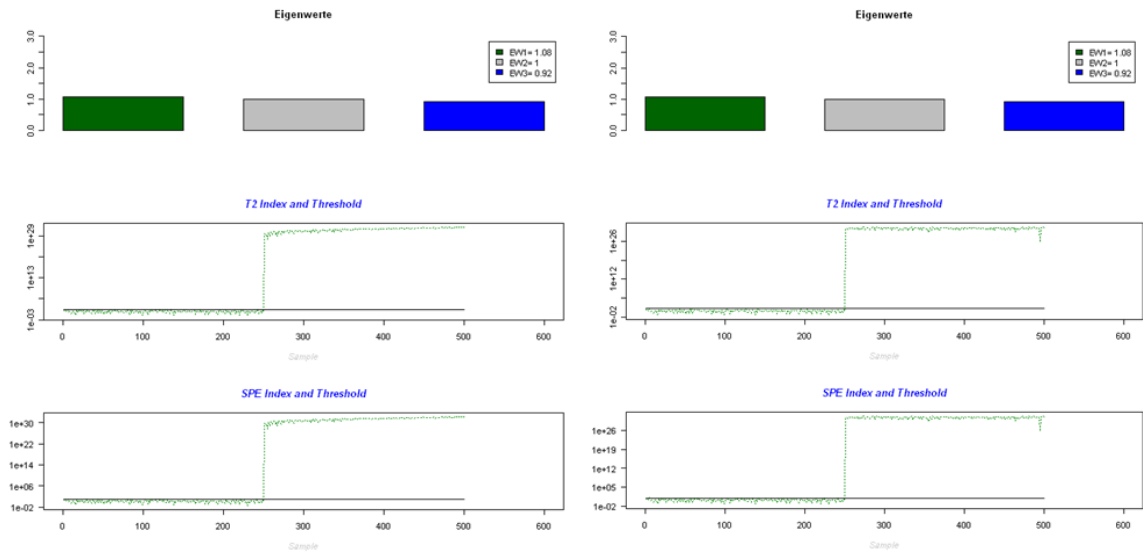


Abb. 4.29: Eigenwerte, SPE - und T^2 -Indizes für multiplikativen Fehler mit normiertem Differenzsignal

Abb. 4.30: Eigenwerte, SPE - und T^2 -Indizes für additiven Fehler mit normiertem Differenzsignal

Auch in diesem Fall wurde trotz korrelierender Signale mithilfe der Differenzbildung sowohl der multiplikative als auch der additive Fehler eindeutig erkannt.

Abschließend kann eingeschätzt werden, dass die Differenzbildung die Detektion multiplikativer und additiver Fehler verbessert. Es wird jedoch empfohlen, eines der korrelierenden Ausgangssignale in die PCA einzubeziehen. Des Weiteren müssen die resultierenden Differenzsignale unbedingt nachträglich normiert werden. Dies ist in den Abläufen des neuen Softwareinformationssystems zu beachten.

4.2.2 Anpassung der Varianz

Eine erfolgreiche Anwendung der PCA-basierten Fehlerdetektion basiert auf der Annahme, dass die Daten einer jeden Prozessvariable normal verteilt sind. Diese Annahme ist jedoch nicht für die PCA selbst entscheidend, sondern für die Anwendung der statistischen Tests, z.B. dem F-Test. Die Abbildung 4.31 zeigt das Diagramm für die Leistung des ersten Antriebsmotors einer Bandantriebsstation über einen Zeitraum von eineinhalb Monaten. Die Einzelwerte sind auf den Mittelwert null und Standardabweichung eins normiert. Die nachfolgenden Abbildungen 4.32 und 4.33 zeigen das dazugehörige Diagramm der Normalverteilung sowie das Histogramm. Eine perfekte Normalverteilung zeigt sich im Normalplot dadurch, dass alle Werte auf der durchgezogenen roten Linie liegen und im Histogramm durch eine Anpassung der Balken an die sogenannten „Glockenkurve“. In der Abbildung 4.31 sieht man deutlich die im Abschnitt 3.5.2 beschriebenen An- und Abfahrvorgänge sowie stationären Zustände. Die nachfolgenden Abbildungen (Diagramm Normalverteilung, Histogramm und

Scatterplot über ausgewählte Daten der Bandantriebsstation) zeigen visuell, dass dieser Datensatz nicht normal verteilt ist.

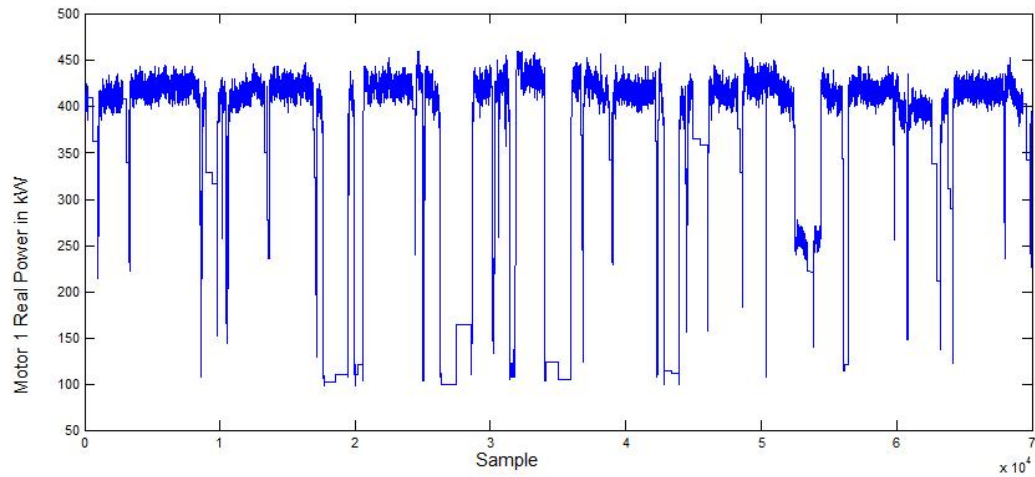


Abb. 4.31: Verlauf für Bandantriebsstation Leistung Motor 1

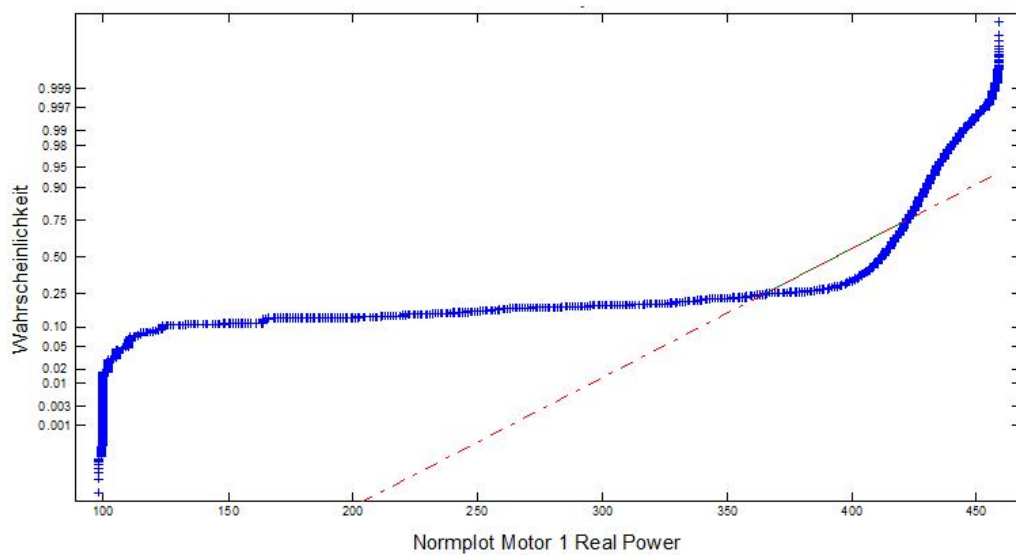


Abb. 4.32: Diagramm Normalverteilung für Bandantriebsstation Leistung Motor 1

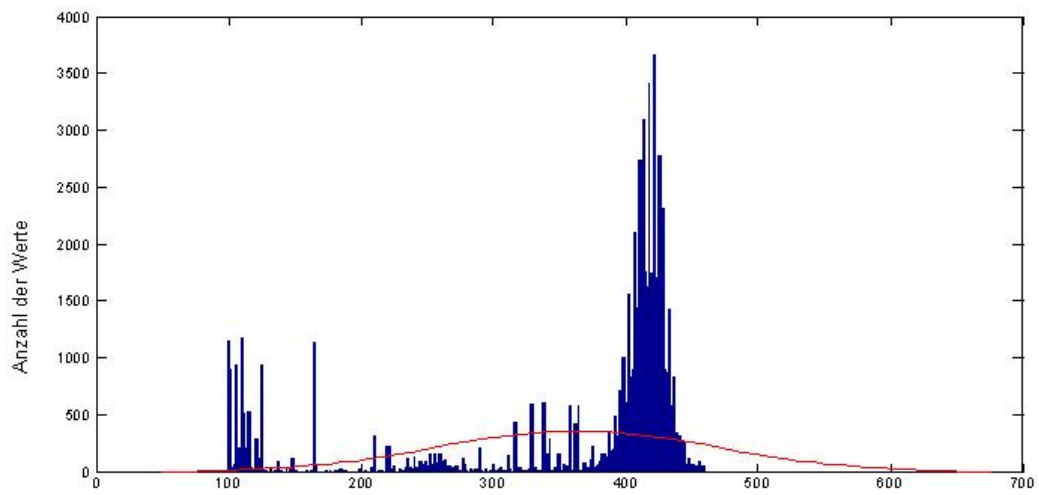


Abb. 4.33: Histogramm für Bandantriebsstation Leistung Motor 1

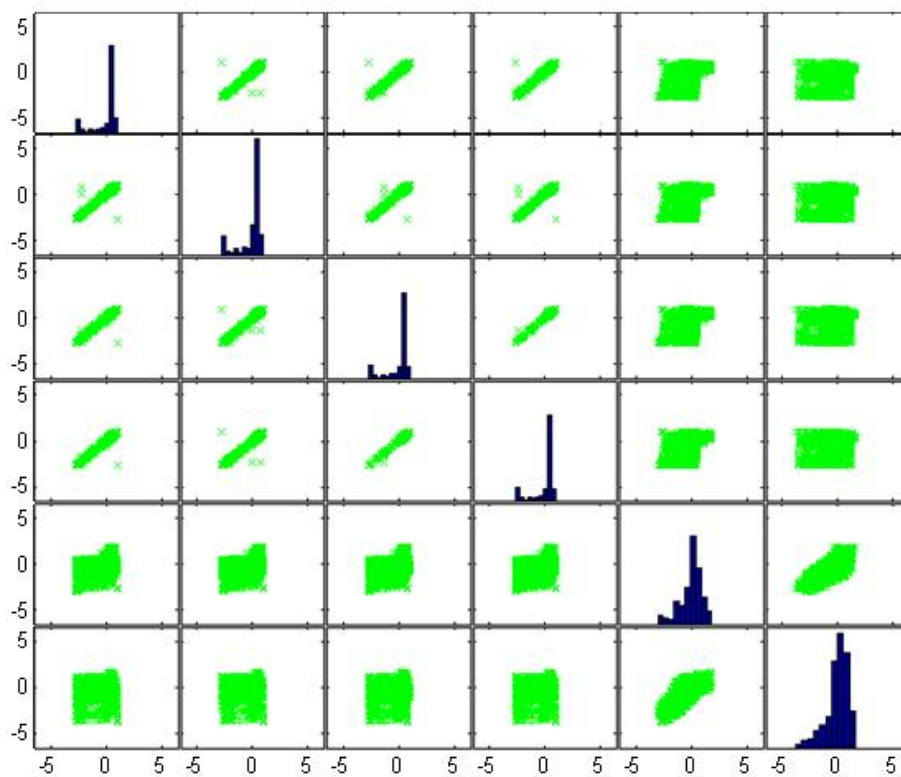


Abb. 4.34: Scatterplot für ausgewählte Daten der Bandantriebsstation vor dem Schneiden

Die Ursachen für die fehlende Normalverteilung liegen beispielsweise in

- unterschiedlichen Arbeitspunkten
- stationären und nicht stationären Zuständen
- Begrenzungen der Signale in der Steuerung oder konstruktionsbedingt
- sowie in Ausreißern oder fehlenden Daten.

Der Mittelwert und die Kovarianzmatrix dienen als statistisches Prozessmodell. Es ist nachgewiesen, dass die datengestützten Methoden PCA (DPCA) sowie GLR nicht effizient funktionieren, wenn diese Voraussetzung nicht erfüllt ist. Wie bereits weiter oben beschrieben, gilt diese Annahme vor allen Dingen für die statistischen Tests. Die Grenzwerte für die Überwachung der berechneten Indizes (T^2 oder SPE bei der PCA) werden in der Annahme einer multivariaten Normalverteilung gebildet. Ist diese aus den verschiedensten Ursachen so nicht gegeben, kommt es selbst im angelernten Anlagen-Normalzustand zu Überschreitungen der Grenzwerte, d.h. zu Fehlalarmen.

Eine mögliche Lösung ist, wenn nur diejenigen Datenabschnitte mit stationären Verhalten berücksichtigt werden. Bei vorhandenem Ein- und Ausschaltsignal der Anlage kann man dieses während der on-line-Überwachung zum Starten und Stoppen des Algorithmus bzw. beim Training zum Herausschneiden der instationären Zustände benutzen. Ist ein solches Signal nicht vorhanden, so kann man einen sogenannten Schnittindex anhand eines Referenzsignals erzeugen, welches eine gute Unterscheidung der stationären von den instationären Zuständen ermöglicht. Die nachfolgende Abbildung zeigt das Referenzsignal, hier die Vibrationen am Motor 1, im Vergleich zum daraus erzeugten Schnittindex. Der Schnittindex, gebildet über eine Signum-Funktion, kann die Werte 0 für instationär und 1 für stationär annehmen.

Die nachfolgenden Diagramme zeigen die über den Schnittindex bereinigten Vibrations- und Leistungsdaten.

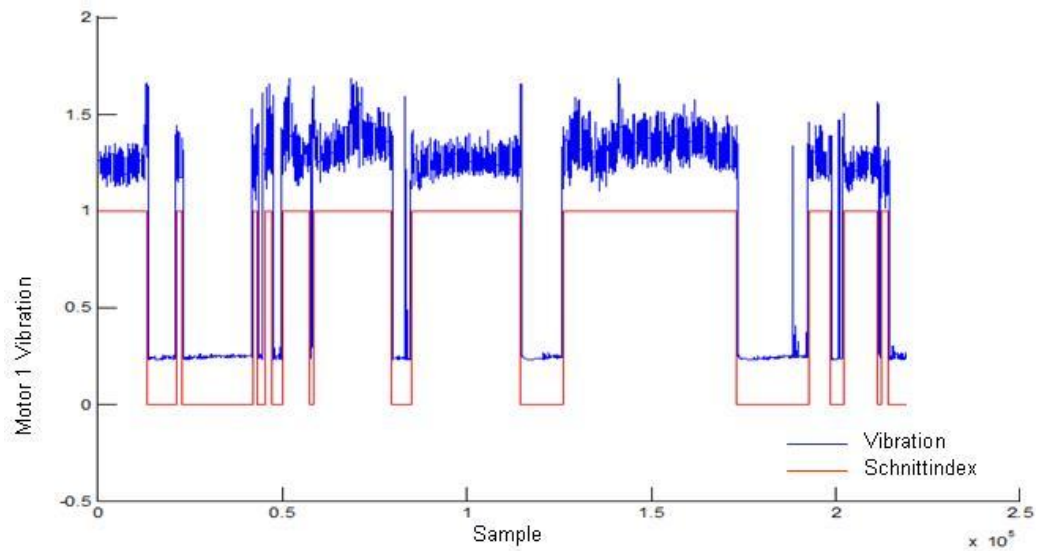


Abb. 4.35: Diagramm Bandantriebsstation Vibration Motor 1 vor dem Schneiden und dem zugehörigen Schnittindex

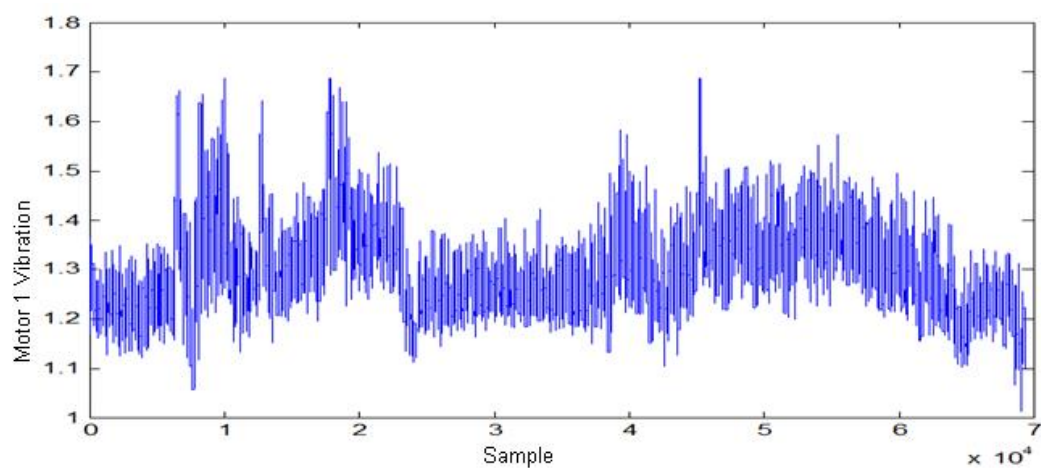


Abb. 4.36: Diagramm Bandantriebsstation Vibration Motor 1 nach dem Schneiden

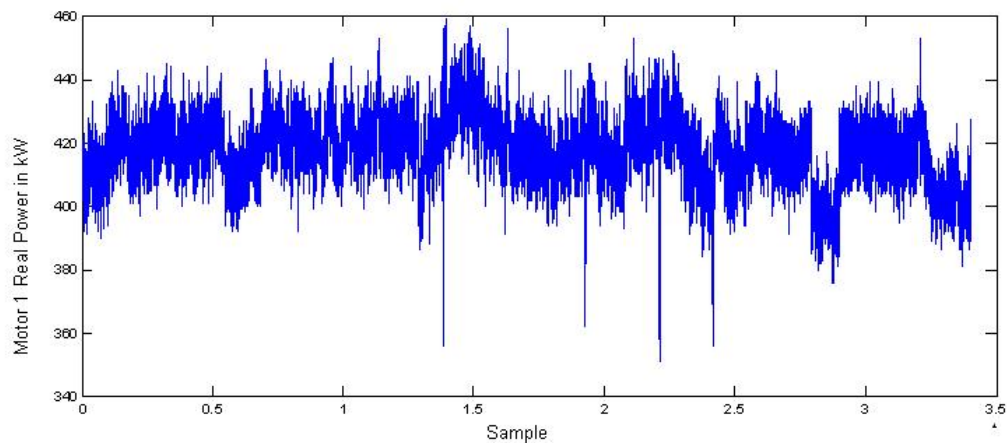


Abb. 4.37: Diagramm Bandantriebsstation Leistung Motor 1 nach dem Schneiden

Das Histogramm und die Normalverteilung für die Leistungsdaten sehen nun folgendermaßen aus:

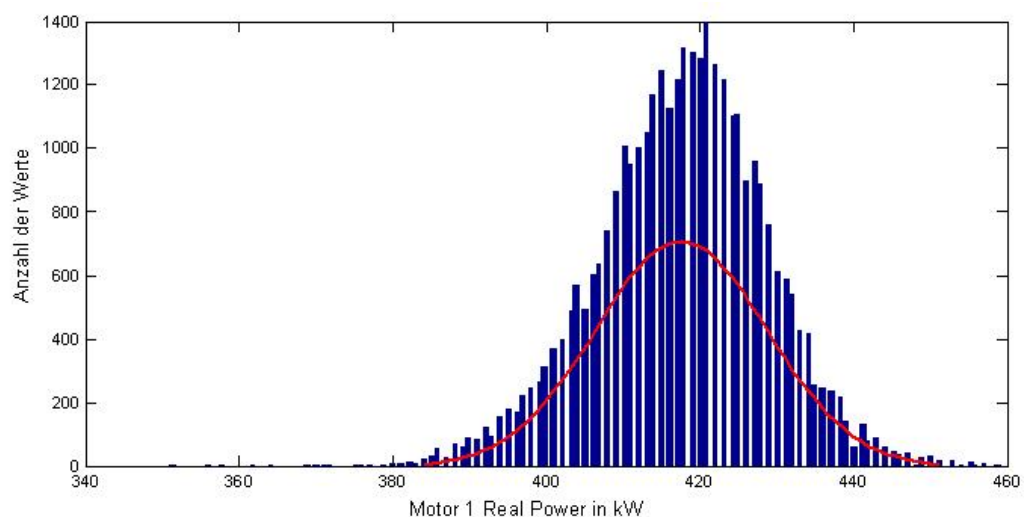


Abb. 4.38: Histogramm für Bandantriebsstation Leistung Motor 1 nach dem Schneiden

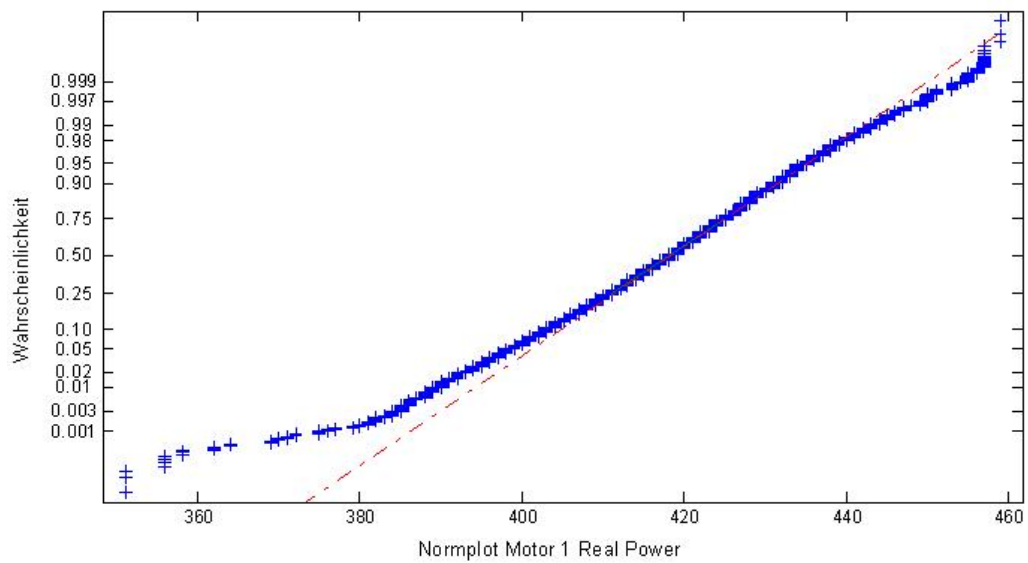


Abb. 4.39: Diagramm Normalverteilung für Bandantriebsstation Leistung Motor 1 nach dem Schneiden

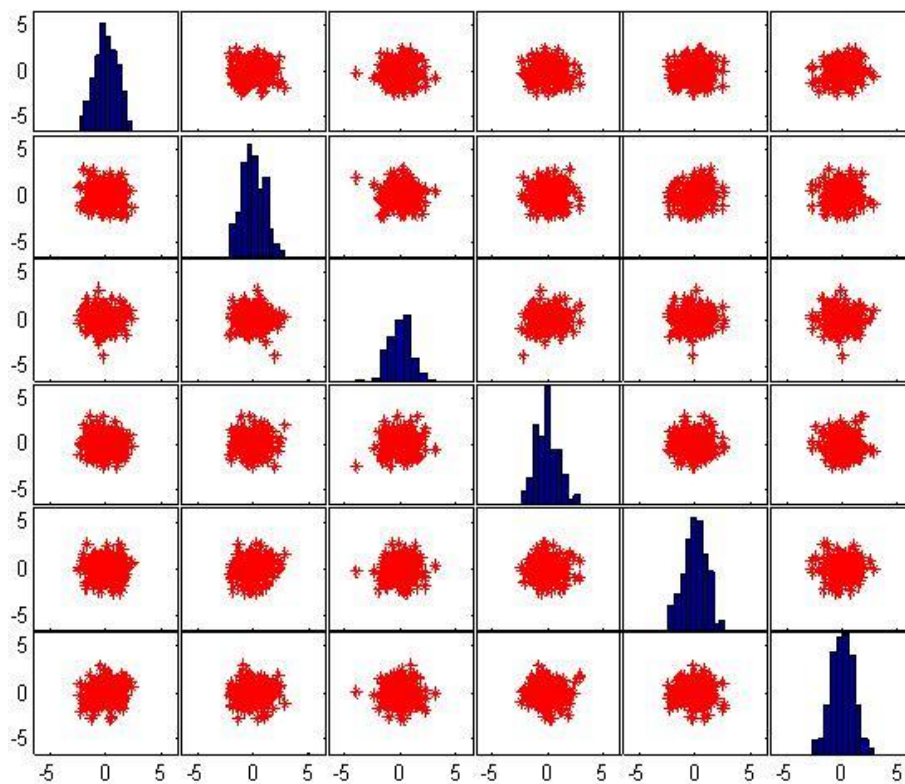


Abb. 4.40: Scatterplot für ausgewählte Daten der Bandantriebsstation nach dem Schneiden

Infolge des Herausschneidens der Daten instationärer Zustände, nähert sich die Verteilung der Daten einer Normalverteilung an. Das zeigt auch der Scatterplot über ausgewählte Daten der Bandantriebsstation nach dem Schneiden (Abbildung 4.40).

Angewendet auf die Prozessdaten vor der Fehlerdetektion hat das Auswirkungen auf die Fehlalarmrate.

Um das zu zeigen, wurden die ungeschnittenen und geschnittenen Daten der Gurtförderanlage untersucht, welche den Normalzustand beschreiben. Nachfolgende Abbildungen zeigen zunächst die Ergebnisse der PCA vor dem Herausschneiden der An- und Abfahrvorgänge.

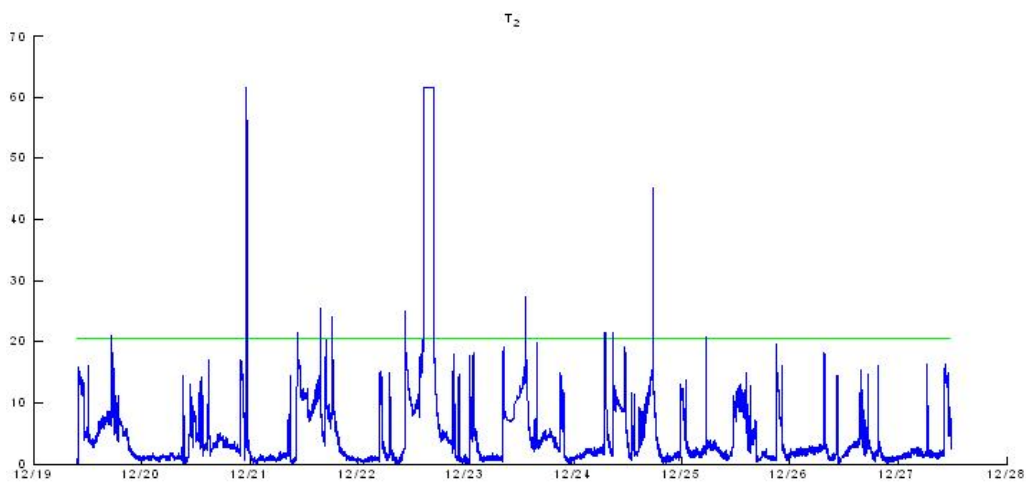


Abb. 4.41: T^2 -Statistik für die Daten des Normalzustandes vor dem Schneiden

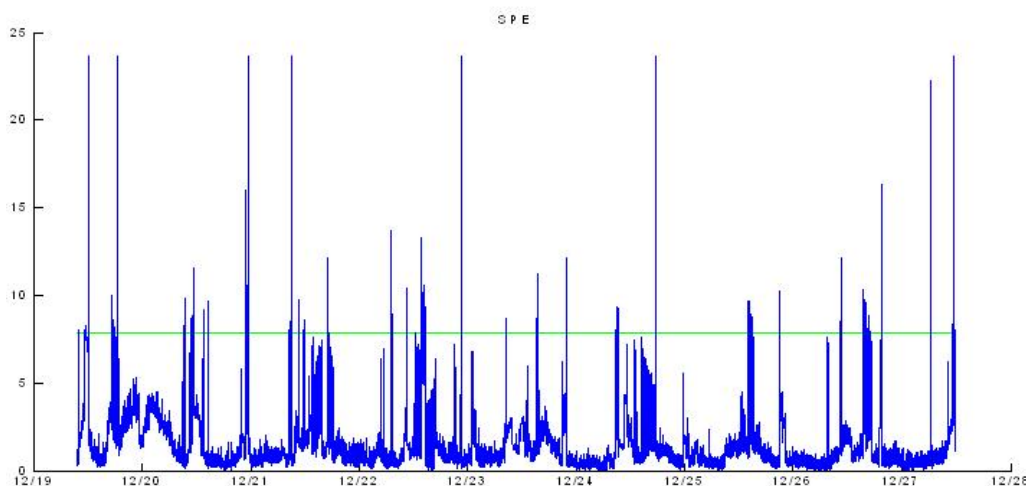


Abb. 4.42: SPE -Statistik für die Daten des Normalzustandes vor dem Schneiden

Die ermittelten Fehlalarmraten betragen für den T^2 -Index $FAR_{T^2} = 1,64\%$ sowie für den SPE -Index $FAR_{SPE} = 1,85\%$.

Das Herausschneiden der An- und Abfahrvorgänge brachte folgende Verbesserungen:

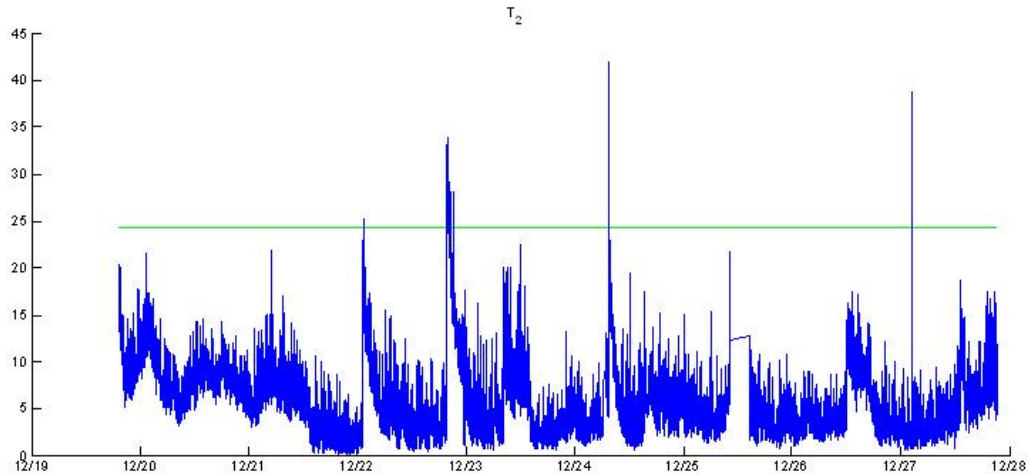


Abb. 4.43: T^2 -Statistik für die Daten des Normalzustandes nach dem Schneiden

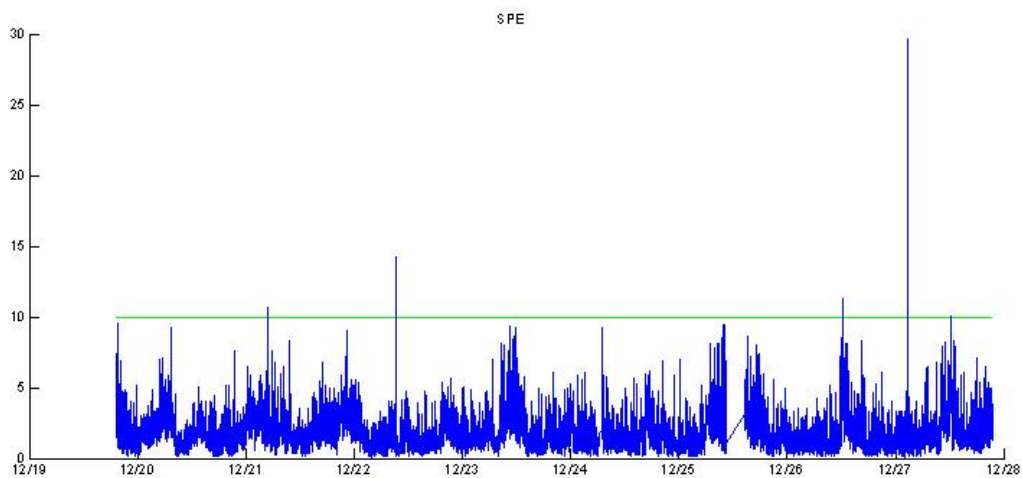


Abb. 4.44: SPE -Statistik für die Daten des Normalzustandes nach dem Schneiden

Nun haben die Fehlalarmraten für den T^2 -Index den Wert von $FAR_{T^2} = 0,158\%$ und für den SPE -Index den Wert von $FAR_{SPE} = 0,021\%$, was einer für den T^2 -Index einer Verbesserung um etwa den Faktor 10 und für den SPE -Index einer Verbesserung um etwa den Faktor 100 entspricht.

Trotz der erzielten Resultate treten immer noch eine Reihe von Grenzwertüberschreitungen auf, welche bei der praktischen Anwendung von datenbasierten Fehlerdetektionsalgorithmen zu für den Anwender wiederkehrenden und irritierenden Fehlalarmen führen würden. Die Ursachen dafür liegen u.a. darin, dass

- Verzögerungen zwischen Ausgangsgrößen und Eingangsgrößen vorliegen, welche systembedingt sind. (So benötigt das Antriebssystem nach dem Einschalten eine bestimmte Zeit, bis es einen stationären Zustand erreicht hat. Beispielsweise braucht die Getriebetemperatur eines Bandantriebssystems etwa zwei Stunden, um von einer Ausgangstemperatur von 48°C zu einem stationären Wert von etwa 63°C zu gelangen.)
- es eine Reihe weiterer Einflüsse und Störgrößen gibt, die zu einer Überschreitung der Indizes führen können, ohne dass es sich dabei um einen Fehlerzustand handelt (Beispiel: das Auftreten einer plötzliche Beladungsänderung)
- nicht alle Wertekombinationen für den fehlerfreien Zustand in den Trainingsdaten enthalten sind.

Aus diesem Grunde entstand die Idee, Standardabweichung und Mittelwert der nicht normalverteilten Daten des Trainings zu bestimmen und mit diesen Werten künstliche normalverteilte Daten zu erzeugen. Die ursprünglichen Trainingsdaten werden anschließend durch die normalverteilten Modelldaten ersetzt, in Eigenwerte und Hauptkomponenten zerlegt und die Grenzwerte (T^2 oder SPE) bestimmt. Da mögliche Ausreißer während des Trainings (Daten über oder unterhalb von 3σ , siehe Abbildung 4.45) auch zum Normal- oder fehlerfreien Zustand gehören, kann für die Varianz der Modelldaten bis zu 3σ der Trainingsdaten angenommen werden. Die Sample-Anzahl der neuen Datenreihen sollte mindestens das 10-fache der Spaltenanzahl betragen, was laut [58] als ausreichend betrachtet wird.

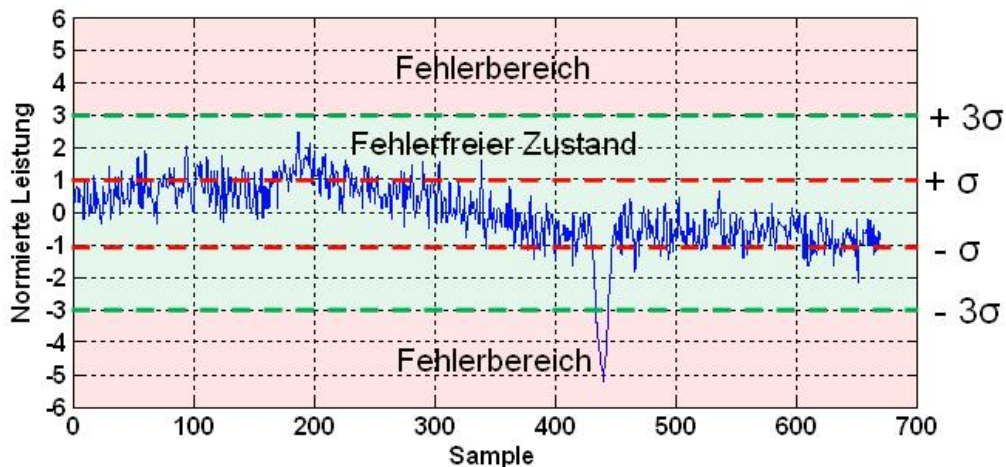


Abb. 4.45: Daten des fehlerfreien Zustandes mit Ausreißer bis zu 3σ

Für das off-line-Training (Design) ergibt sich nun folgender Ablauf:

1. Berechnung der Erwartungswerte und der Standardabweichungen aus den Trainingsdaten

2. Erzeugung multinormalverteilter Daten mit den Erwartungswerten und den Standardabweichungen aus den Trainingsdaten und nachfolgend angepasster Varianz
3. Berechnung des Schwellenwertes nach Gleichung 3.21 für die Überwachung mit der PCA oder 3.72 für die Überwachung mithilfe der GLR-Methode
4. Berechnung der Erwartungswerte, Standardabweichungen und der Kovarianzmatrix aus den Modelldaten für die nachfolgende Indexberechnung.

Die nachfolgenden Abbildungen zeigen die veränderten Abläufe beim Training sowohl für das PCA- und GLR-Fehlerdetektionsverfahren.

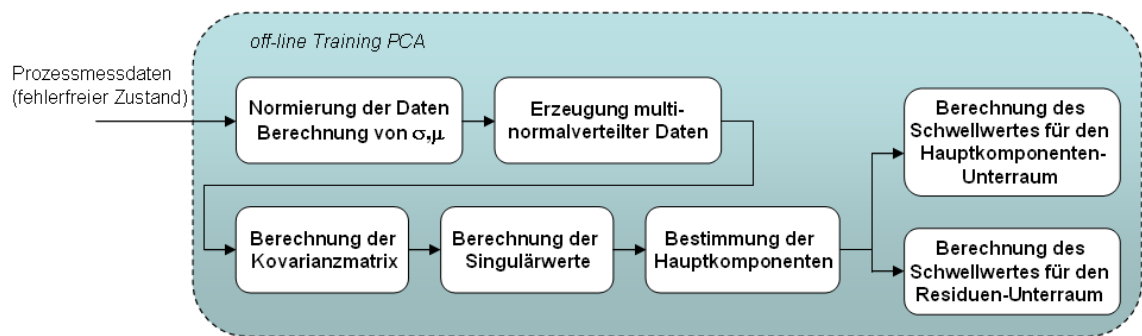


Abb. 4.46: off-line-Training PCA zur Ermittlung der angepassten Kovarianzmatrix

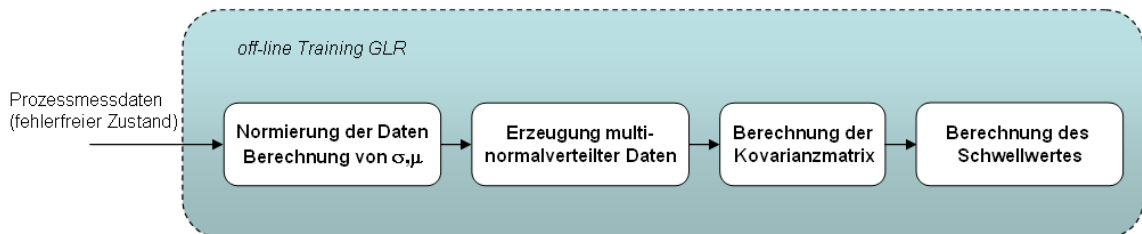


Abb. 4.47: off-line-Training GLR zur Ermittlung der angepassten Kovarianzmatrix

Die Prozessschritte für die on-line-Überwachung verändern sich ebenfalls:

1. Normierung der Testdaten auf den Erwartungswert und die Varianz der Trainingsmodelldaten,
2. Residuengenerierung auf Basis der Modell-Kovarianzmatrix bzw. der daraus ermittelten Hauptkomponenten,
3. Residuenauswertung nach folgender Logik,

$$g(z) = \begin{cases} 1 & \text{falls } T^2 > J_{th,F} \Rightarrow \text{Fehler} \\ 0 & \text{falls } T^2 \leq J_{th,F} \Rightarrow \text{Normalzustand} \end{cases} \quad (4.22)$$

Grafisch veranschaulicht sieht der Ablauf nun so aus:

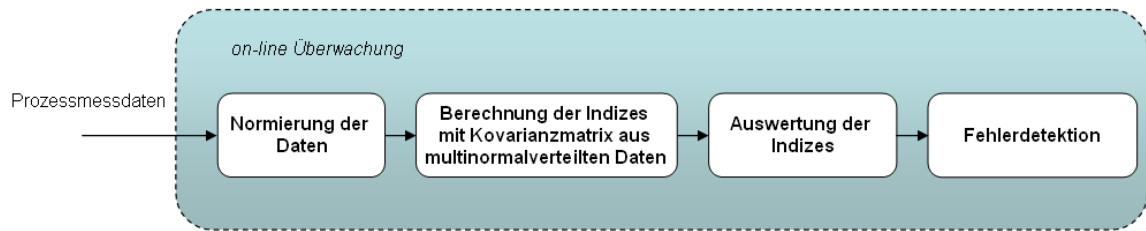


Abb. 4.48: on-line-Überwachung (GLR und PCA) auf Basis der angepassten Kovarianzmatrix

Die beiden nachfolgenden Abbildungen zeigen die PCA-Indizes und -Schwellwerte für die Daten des Normalzustandes, wobei die Normierung und die Bildung der Kovarianzmatrix auf der Basis der künstlich generierten Modelldaten (Varianz: 3σ) erfolgte.

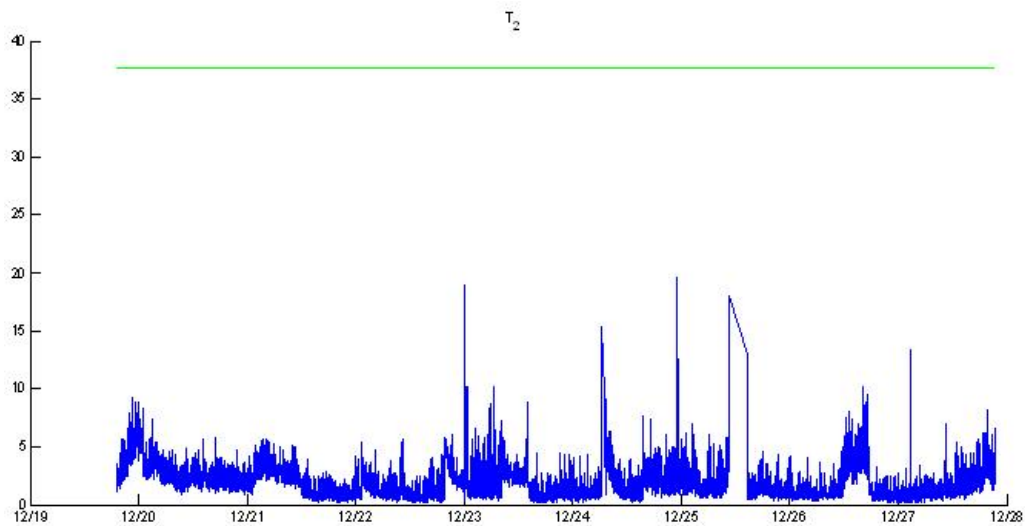


Abb. 4.49: T^2 -Statistik für die Daten des Normalzustandes nach dem Schneiden

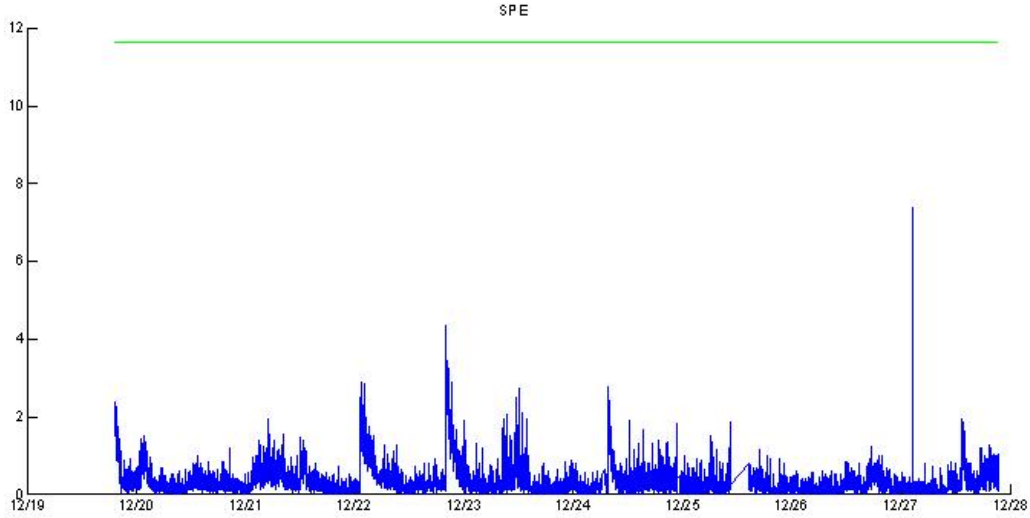


Abb. 4.50: *SPE*-Statistik für die Daten des Normalzustandes nach dem Schneiden

Die so ermittelten Grenzwerte sind robuster gegenüber Einflussgrößen, welche den stationären Zustand beeinflussen aber keinen Fehler darstellen. Fehlalarme im Normalzustand treten nicht mehr bzw. kaum noch auf, Anlagen- und Prozessfehler werden aber ausreichend sicher erkannt.

4.3 Analyse der Zusammenhänge zwischen PCA und GLR

Der Kern der PCA-Technik besteht in der Zerlegung des normalisierten Beobachtungsraumes in den sogenannten Unterraum der Hauptkomponenten P_{pc} und dem Residuum-Unterraum P_{res} . Da sich ein Fehler in einem der beiden Unterräume zeigen kann, erfolgt in der Regel die Fehlerüberwachung auch in beiden Räumen. Wie in Kapitel 4 beschrieben werden dazu die Prozessdaten z auf P_{pc} und P_{res} projiziert. In [16] wird darauf hingewiesen, dass die Fehlerstatistik im Residuum-Unterraum oftmals die besseren Ergebnisse liefert als die im Hauptkomponenten-Unterraum. Die ursprüngliche Idee, nach der die PCA auf die Reduzierung der Dimensionen unter Beibehaltung der maximalen Varianz zielt, bringt unter diesen Umständen keine Vorteile.

Andererseits führt die in dem Abschnitt 3.3.5 beschriebene GLR-Technik mit der Teststatistik laut Gleichung 3.69 und der Testvorschrift (3.72) ebenfalls zu guten Ergebnissen bei der Fehlerdetektion. Vergleicht man den T^2 -Index der PCA mit der GLR-Teststatistik, so stellt man fest, dass die GLR-Teststatistik für $N = 1$ dem T^2 -Index für den Bereich der Hauptkomponenten gleicht. Für die Fehlerdetektion im Residuum-Unterraum kann die sogenannte Hawkins Statistik mit $T_H^2 = z^T P_{res} \Lambda^{-1} P_{res}^T z$ verwendet werden. Sind die Eigenwerte des Unterraumes aber sehr

klein (nahe null), so kann das zu einer schlechten Konditionierung von Λ^{-1} und damit zu schlechten Ergebnissen bei der Fehlerdetektion führen. Um die numerischen Probleme zu vermeiden, wird in [16] die folgende Erweiterung von Hawkin's Statistik vorgeschlagen:

$$T^2 = z^T P_{res} \Theta P_{res}^T z \quad (4.23)$$

mit

$$\Theta = diag(\frac{\sigma_m^2}{\sigma_{l+1}^2}, \dots, \frac{\sigma_m^2}{\sigma_{m-1}^2}, 1) \quad (4.24)$$

Der Schwellwert kann über $J_{th,res} = \sigma_m^2 X_\alpha^2(m-l)$ berechnet werden. Diese Statistik ist ebenfalls X^2 verteilt.

Kombinierte Indizes

Unter dem Gesichtspunkt einer einfach anwendbaren, interpretierbaren Fehlerdetektion ist es von Vorteil, wenn man nur einen Index auswerten muss, anstatt zwei, wie bei der PCA. Qin schlägt deshalb in [51] und [52] die Verwendung eines kombinierten Indexes aus T^2 und SPE vor. Der kombinierte Index kann allgemein folgendermaßen formuliert werden:

$$T_{Comb,Qin}^2 = \beta_1 T_{pc}^2 + \beta_2 T_{res}^2. \quad (4.25)$$

Die Werte β_1 und β_2 sind größer 0 und ergeben sich aus den berechneten zugehörigen Schwellwerten.

Verwendet man nun für den Residuum-Bereich statt des SPE-Indexes den Index laut Gleichung 4.23, so ergibt sich folgender kombinierter Index:

$$T_{Comb}^2 = z^T P \bar{\Theta} P^T z, \bar{\Theta} = diag(\frac{\sigma_m^2}{\sigma_1^2}, \dots, \frac{\sigma_m^2}{\sigma_{m-1}^2}, 1) = \sigma_m^2 (T^2 + T_H^2), \quad (4.26)$$

vorausgesetzt, dass $P^T z \sim \mathcal{N}(0, \Lambda)$ und $z^T P \Lambda^{-1} P^T z$ ist $\chi_{m,\alpha}^2$ verteilt, mit m Freiheitsgraden. Der zugehörige Schwellwert für ein vorgegebenes α ist:

$$J_{Th,Comb} = \sigma_m^2 \chi_\alpha^2(m), \quad (4.27)$$

Die Statistik T_{Comb}^2 kann als eine gewichtete quadratische Form der projizierten Beobachtungen $P^T z$ betrachtet werden. Beachtenswert ist, dass diese Statistik Werte in Richtung der stärkeren Varianzanteile geringer wichtet, als Werte in Richtung einer schwächeren Varianz. Ist der Wert von σ_m^2 sehr klein, also nahe null, so führen bereits kleine Ausschläge in diese Richtung zu einer maximalen Wichtung damit zur Fehleranzeige.

Es ist offensichtlich, dass T_{Comb}^2 und der damit verbundene Schwellwert $J_{Th,Comb}$ aus 4.26 und 4.27 der im GLR-Statistik und -Schwellwert aus dem Abschnitt 3.3.5 entsprechen, wobei bei der GLR keine Eigenwertzerlegung und auch keine Projektionen

auf die nach dem Varianzanteil geordneten Eigenvektoren erfolgen. Da mit der im Abschnitt 3.3.5 beschriebenen multivariaten GLR-Methode ebenfalls optimale Ergebnisse bei der Fehlerdetektion erzielt werden können, ist der Schritt der Eigenwertzerlegung und Projektion auf alle Eigenvektoren, wie bei der PCA, offensichtlich überflüssig. Des Weiteren hat die multivariate GLR-Methode den Vorteil, dass bei einer entsprechend großen Wahl von N die Historie von N -Werten bei der Berechnung des Indexes beachtet wird, wodurch die Robustheit gegenüber Ausreißern steigt.

4.4 Adaptive GLR-Methode

Steht man vor der im Abschnitt 3.3.3 beschriebenen Problematik großer Unsicherheiten im industriellen Umfeld, in dem sich Prozessparameter über die Zeit ändern, ohne dass diese Änderungen Ausdruck für einen Fehlerzustand sind, so sind die adaptiven und rekursiven Techniken effiziente und weitverbreitete Methoden, die uns erlauben, über die on-line-Aktualisierung des Prozess- und Anlagen-Modells die (unsicheren) Änderungen im Anlagenprozess zu bewältigen. Für die PCA wurde ein solcher rekursiver Algorithmus im Abschnitt 3.3.3 vorgestellt. Soll unter diesen Prozessbedingungen die GLR-Methode zur Anwendung kommen, so kann auch hier der Einsatz adaptiver und rekursiver Techniken die Robustheit der Überwachungsmethode verbessern. In diesem Abschnitt soll eine solche adaptive GLR-Technik vorgestellt werden. Bei der im Abschnitt 3.3.3 vorgestellten on-line-Überwachung mit der RPCA wurden aufgrund des erhöhten Rechenaufwandes bei der Eigenwertzerlegung mit jedem neuen Datenvektor nicht die Kovarianzmatrix, sondern direkt die Eigenwerte und Eigenvektoren angepasst. Zum statistischen Modell, welches bei der GLR-Methode zur Anwendung kommt, gehören der Mittelwert und die Kovarianzmatrix aus dem off-line-Training. Da bei der GLR keine Eigenwert- oder Singulärwertzerlegung notwendig ist, kann man im Gegensatz zur RPCA bei der on-line-Überwachung mit der GLR-Methode nun direkt die Kovarianzmatrix sowie den Mittelwert anpassen.

4.4.1 Partielle adaptive GLR-Methode

Eine der häufig angewendeten Varianten, schleichende Änderungen in den Prozessdaten, die keinen Fehler darstellen, nicht in die datenbasierten Berechnungen zur Fehlerdetektion einfließen zu lassen, ist die Anpassung von Mittelwert und Standardabweichung während der Normierung und Mittelzentrierung der Daten. Die von Spitzlsperger et al. [65] vorgestellte Methode des partiellen adaptiven Updates für die PCA kann auch für die GLR angewendet werden. Für den neuen Mittelwert und die Standardabweichung bedeutet das:

$$\boldsymbol{\mu}_{k+1} = (1 - \alpha)\mathbf{x}_{k+1} + \alpha\boldsymbol{\mu}_k. \quad (4.28)$$

$$\sigma_{k+1} = \sqrt{\sigma_k^2 \times \frac{m-2}{m-1} + \frac{1}{m} (\mathbf{x}_{k+1} - \boldsymbol{\mu}_k)^2} . \quad (4.29)$$

Die Werte von m und α sind anpassbar. Niedrige Werte von m führen zu einer höheren Wichtung der jüngsten Daten, da der zweite Term unter der Quadratwurzel in 4.29 dominanter wird.

4.4.2 Adaptive GLR-Methode mit Vergessensfaktor

Ausgangspunkte sind die Kovarianzmatrix zum Zeitpunkt k \mathbf{S}_k , der neu eingetroffene Datenvektor zum Zeitpunkt $k+1$ \mathbf{x}_{k+1} und der Vergessensfaktor α . Darüber hinaus kann man unterscheiden, ob bereits nach jedem neuen Datensatz (Datenvektor) ein Update durchgeführt wird oder erst nach mehreren neuen Datensätzen, einem ganzen Datenblock.

Update nach einem Datensatz

Den neuen Mittelwert berechnet sich nach der bereits in der partiellen adaptiven GLR-Methode aufgeführten Formel 4.28:

Danach erfolgt die Mittelwertzentrierung des neuen Datensatzes mit dem angepassten Mittelwert.

$$\mathbf{z}_{k+1} = \mathbf{x}_{k+1} - \boldsymbol{\mu}_{k+1} \quad (4.30)$$

Anschließend ermittelt man die neue Kovarianzmatrix.

$$\mathbf{S}_{k+1} = \alpha \mathbf{S}_k + (1 - \alpha) \mathbf{z}_{k+1}^T \mathbf{z}_{k+1} \quad (4.31)$$

Update nach einem neuen Datenblock

Wenn sich die Änderung der Prozessparameter sehr langsam im Vergleich zu der Abtastrate der Daten vollzieht, dann kann man das Update auch nach einem Block von neuen Daten durchführen, anstatt zu jedem einzelnen neuen Datensatz. Das hat den Vorteil, dass erstens deutlich weniger Rechenaufwand benötigt wird und zweitens immer ein ganzer Block von Daten für das neue Modell herangezogen wird und damit das Risiko der Falschalarmierung sinkt. Die rekursive, blockweise Berechnung für den neuen Mittelwert und für die Kovarianz sieht nun nach [8] folgendermaßen aus:

$$\boldsymbol{\mu}_{k+1} = (1 - \alpha) \mathbf{X}_{B,k+1}^T \mathbf{1}_{B,k+1} + \alpha \boldsymbol{\mu}_k \quad (4.32)$$

$$\mathbf{S}_{k+1} = \alpha \mathbf{S}_k + (1 - \alpha) \mathbf{Z}_{B,k+1}^T \mathbf{Z}_{B,k+1}, \quad (4.33)$$

wobei $X_{B,i} \in \mathbb{R}^{N_i \times m}$ der i -te Datenblock, m die Anzahl der Variablen und $\mathbf{1}_{B,i} \in \mathbb{R}^{N_i}$ sind. Laut Choi et al. [8] kann der Vergessensfaktor auch mit jedem neuen Schritt neu berechnet werden. Die entsprechenden Algorithmen sind in [8] zu finden.

Mit dem angepassten Mittelwert und der angepassten Kovarianzmatrix erfolgt nun die Berechnung des T^2 -Indexes und der Vergleich mit dem Schwellwert laut Gleichung 3.73. Wie auch bei der rekursiven PCA sollte beachtet werden, dass die Adaption des Modells nach einem Fehler stoppt, da sonst der Fehler in das Modell integriert wird.

4.4.3 Adaptive GLR-Methode mit einem gleitenden Fenster

Bei Prozessen, die sich sehr langsam ändern und eine enge Abtastrate besitzen, ist es nicht ganz einfach, den Vergessensfaktor optimal zu wählen. Beispielsweise können sich Temperaturunterschiede, bedingt durch Schwankungen zwischen Tag und Nacht oder durch Schwankungen über die Jahreszeiten, sehr langsam auf das Prozessverhalten auswirken. Arbeitet man aber mit einem gleitenden Fenster, kann über die Fenstergröße (Anzahl der Samples innerhalb des Fensters) ein zeitlicher Bezug zu der Änderung von einzelnen Prozessparametern hergestellt werden. Aus diesem Grunde werden Teile des von He und Yang in [28] vorgestellten Moving-Window-PCA (MWPCA) für eine Moving-Window-GLR-Methode verwendet.

Ausgangspunkt ist die Matrix $\mathbf{X}_k^0 \in \mathbb{R}^{L \times m}$ mit L Samples und m Variablen (Spalten). L bestimmt somit die Fenstergröße. Aus dieser Matrix kann $\boldsymbol{\mu}_k \in \mathbb{R}^{1 \times m}$, $\boldsymbol{\sigma}_k \in \mathbb{R}^{1 \times m}$ die normalisierte Datenmatrix $\mathbf{X}_k \in \mathbb{R}^{L \times m}$ und die Kovarianzmatrix $\mathbf{R}_k \in \mathbb{R}^{m \times m}$ bestimmt werden. Die Aufgabe besteht nun in der Neuberechnung von $\boldsymbol{\mu}_{k+1}$, \mathbf{X}_{k+1} und \mathbf{R}_{k+1} , wenn der nächste Datenblock \mathbf{x}_{new} eintrifft. Wie in [28] beschrieben ist bei der rekursiven Berechnung der Kovarianzmatrix die Anpassung von $\boldsymbol{\Sigma}$ nicht unbedingt notwendig, weshalb zur Skalierung $\boldsymbol{\Sigma}_k$ verwendet wird.

Zunächst erfolgen das Entfernen des ältesten Datenblockes aus dem Datenfenster und die Berechnung der entsprechenden Parameter.

$$\tilde{\boldsymbol{\mu}} = \frac{1}{L - n_{k+1} - 1} \left(L\boldsymbol{\mu}_k - \mathbf{I}_{k+1}^T \mathbf{x}_{old}^0 \right) \quad (4.34)$$

Die Matrix $\tilde{\mathbf{X}}^0$ ist nun zu normieren, woraus sich ergibt:

$$\tilde{\mathbf{X}} = \left(\tilde{\mathbf{X}}^0 - \mathbf{I}_{k+1} \tilde{\boldsymbol{\mu}} \right) \boldsymbol{\Sigma}_k^{-1} \quad (4.35)$$

$$\begin{aligned} \tilde{\mathbf{R}} = \frac{L - 1}{L - n_{k+1} - 1} \mathbf{R}_k - \frac{L - n_{k+1}}{L - n_{k+1} - 1} \boldsymbol{\Sigma}_k^{-1} \Delta \tilde{\boldsymbol{\mu}}^T \Delta \tilde{\boldsymbol{\mu}} \boldsymbol{\Sigma}_k^{-1} - \\ \frac{1}{L - n_{k+1} - 1} \mathbf{x}_{old}^T \mathbf{x}_{old} \end{aligned} \quad (4.36)$$

An dieser Stelle sind $\Delta\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_k - \tilde{\boldsymbol{\mu}}$ und $\mathbf{I}_{k+1} = [1, 1, \dots, 1, 1]^T \in \mathbb{R}^{n_{k+1}}$.

Ist ein neuer Datenblock $\mathbf{x}_{new}^0 \in \mathbb{R}^{n_{k+1}}$ zum Zeitpunkt $k+1$ verfügbar, wird dieser in die Fenstermatrix integriert, sodass die Matrix $\mathbf{X}_{k+1}^0 \in \mathbb{R}^{L \times m}$ entsteht. Anschließend berechnet man die neuen Mittelwerte $\boldsymbol{\mu}_{k+1}$ und die neue Kovarianzmatrix \mathbf{R}_{k+1}

$$\boldsymbol{\mu}_{k+1} = \frac{1}{L-1} \left(L\boldsymbol{\mu}_k + \mathbf{I}_{n_{k+1}}^T \mathbf{x}_{new}^0 \right) \quad (4.37)$$

$$\begin{aligned} \mathbf{R}_{k+1} = \frac{L - n_{k+1} - 1}{L - 1} \tilde{\mathbf{R}} + \frac{L - n_{k+1}}{L - 1} \boldsymbol{\Sigma}_k^{-1} \Delta\boldsymbol{\mu}_{k+1} \Delta\boldsymbol{\mu}_{k+1}^T \boldsymbol{\Sigma}_k^{-1} - \\ \frac{1}{L - 1} \mathbf{x}_{new}^T \mathbf{x}_{new}, \end{aligned} \quad (4.38)$$

wobei $\Delta\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_{k+1} - \tilde{\boldsymbol{\mu}}$ ist.

Nach dem Einsetzen von Gleichung 4.37 in Gleichung 4.38 erhält man:

$$\begin{aligned} \mathbf{R}_{k+1} = \mathbf{R}_k - \frac{L - n_{k+1}}{L - 1} \boldsymbol{\Sigma}_k^{-1} \Delta\tilde{\boldsymbol{\mu}} \Delta\tilde{\boldsymbol{\mu}}^T \boldsymbol{\Sigma}_k^{-1} - \frac{1}{L - 1} \mathbf{x}_{old}^T \mathbf{x}_{old} + \\ \frac{L - n_{k+1}}{L - 1} \boldsymbol{\Sigma}_k^{-1} \Delta\boldsymbol{\mu}_{k+1} \Delta\boldsymbol{\mu}_{k+1}^T \boldsymbol{\Sigma}_k^{-1} + \frac{1}{L - 1} \mathbf{x}_{new}^T \mathbf{x}_{new}. \end{aligned} \quad (4.39)$$

Wie bei der von Choi et al. in [8] vorgestellten Methode zur adaptiven Berechnung des Vergessensfaktors kann für die Moving-Window-Methode die Fenstergröße L adaptiv berechnet werden. Die entsprechenden Algorithmen sind in [28] aufgelistet.

Wie gehabt, erfolgt nun mit dem angepassten Mittelwert und mit der angepassten Kovarianzmatrix die Berechnung des T^2 -Indexes und der Vergleich mit dem Schwellwert laut Gleichung 3.73.

4.5 Zusammenfassung

Die Ergebnisse dieses Kapitels können folgendermaßen zusammengefasst werden:

- Die Datenaufbereitung vor der Anwendung der Fehlerdetektionsmethoden hat eine entscheidende Bedeutung für die Ergebnisse der Fehlerdetektion mittels der Methoden PCA und GLR, da Voraussetzungen, welche die Daten erfüllen müssen oftmals nicht gegeben sind. An- und Abfahrvorgänge, Störsignale und andere Aspekte beeinflussen die Verteilung der Daten und die dafür entscheidenden Kenngrößen.
- Die Bildung von Differenzsignalen aus stark korrelierenden Signalen verbessert die Detektierbarkeit multiplikativer und additiver Fehler mithilfe der PCA. Die nachträgliche Normierung der Differenzsignale ist unbedingt notwendig.

- Es konnte gezeigt werden, dass es möglich ist, robuste statistische Grenzwerte zur Fehlerdetektion mithilfe der Methoden PCA oder GLR auf der Basis künstlicher Modelldaten zu erzeugen. Das Generieren der Modelldaten erfolgt auf der Basis von Erwartungswert und Standardabweichung der Trainingsdaten.
- Die Methode des Generalized-Likelihood-Quotienten basierend auf einem Maximum-Likelihood-Schätzer für den Erwartungswert führt zur gleichen Teststatistik, welche bei der PCA für die Hauptkomponenten zur Anwendung kommt. Da viele Fehler deutlich besser im Residuum-Unterraum detektiert werden können, somit eine Einsparung von CPU-Kosten nicht gegeben ist, kann man den Schritt der Eigenwertzerlegung weglassen. Das führt in letzter Konsequenz zur Anwendung der GLR-Technik.
- Die Fehlerdetektion mit der Methode der GLR konnte durch drei adaptive Verfahren zur Indexberechnung ergänzt werden. Da die Arbeit mit einem Vergessensfaktor bei sich langsam ändernden Prozessen nicht unproblematisch ist, wird die adaptive GLR mit einem gleitenden Fenster empfohlen.

Die in diesem Kapitel vorgestellten Ansätze und veränderten Algorithmen zur Datenaufbereitung, Detektion und Adaption sollen die Arbeit der datenbasierten Methoden zur Fehlerdetektion verbessern oder eine Fehlerdetektion überhaupt erst ermöglichen.

5 Entwurf des neuen Softwareinformationssystems

5.1 Motivation

Wie bereits bei der Konkretisierung der Aufgabenstellung beschrieben, sollen die in den Kapiteln 3 und 4 untersuchten und weiterentwickelten datenbasierten multivariaten Methoden in das Asset-Management-System zedas®asset der Firma PC-Soft GmbH integriert werden. Das neue Softwaremodul ermöglicht die Fehlerdetektion und -identifikation auf der Basis leistungsfähiger datenbasierter, statistischer Algorithmen und ergänzt die vorhandenen Diagnosemöglichkeiten von zedas®asset.

Ein entscheidender und grundlegender Schritt bei der Entwicklung einer neuen Software ist der Entwurf der Architektur. Hierbei wird eine Evaluierung des Systems zu einem möglichst frühen Zeitpunkt im Entwurfsprozess angestrebt. Das Softwareinformationssystem soll sich als Modul in andere bestehende Softwaresysteme integrieren, bzw. Schnittstellen zu vorgelagerten Steuerungs- und Leitsystemen und nachgeordneten ERP-Systemen besitzen. Ein wichtiger Punkt ist hierbei die Beherrschung der exponentiell wachsenden Komplexität solcher Systeme, deren Entwurf sich erheblich vom früheren klassischen Software-Entwurf unterscheidet. Das versucht man zu erreichen, indem man kostengünstige Standardsoftwarekomponenten einsetzt. Was im Bereich der Computerhardware bereits gelungen ist, setzt man nun auch bei der Entwicklung von Softwarekomponenten um. Die Softwarekomponenten müssen auf einem hohen Abstraktionsniveau zur Verfügung gestellt werden, um eine individuelle Anpassung der Software an die teilweise beschränkten Systemressourcen, wie Speicherplatz und Rechenleistung sowie an die verschiedenen Aufgaben zu ermöglichen.

Kern des neuen Softwareinformationssystems sind die datenbasierten multivariaten Methoden, welche zu Anlagendiagnose modifiziert wurden. Nachfolgende Abbildung 5.1 soll den prinzipiellen Ablauf verdeutlichen.

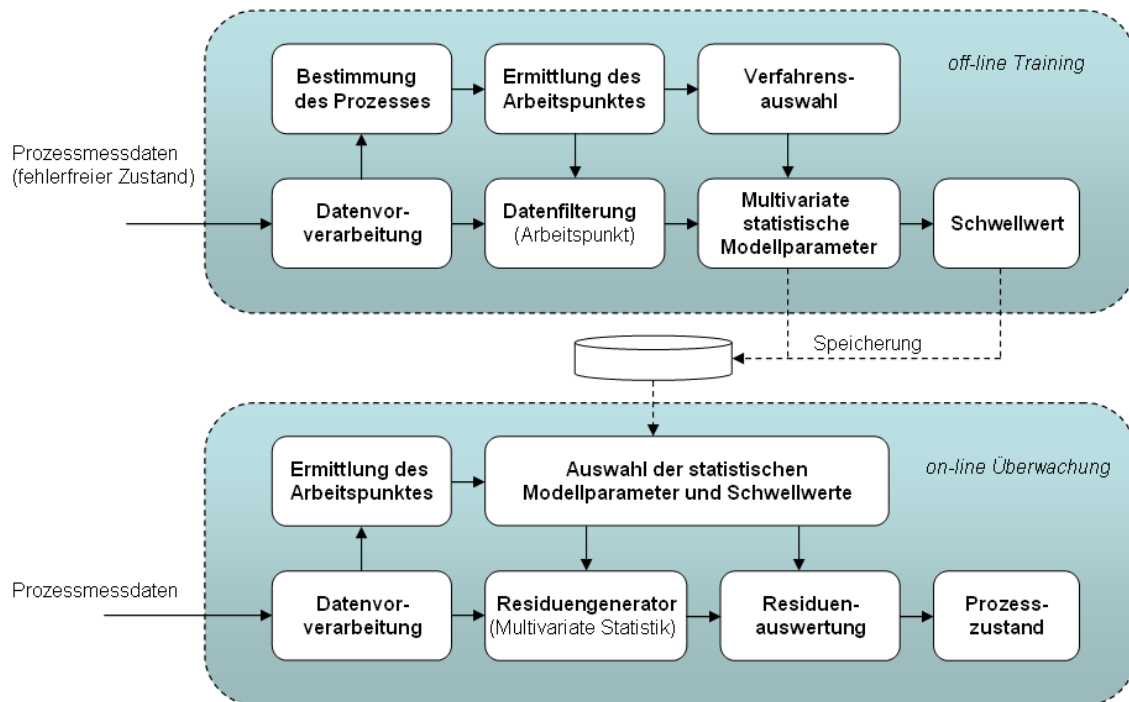


Abb. 5.1: Fehlerdetektionssystem auf der Basis multivariater Methoden

Die Umsetzung der entwickelten Algorithmen in Software erfolgt mithilfe eines verfügbaren Numerik-Frameworks. Dieses Framework soll die grundlegenden geforderten mathematischen Operationen unterstützen, sich in die vorgeschlagene Architektur einordnen sowie die Anforderungen an Geschwindigkeit, Systemressourcen usw. erfüllen. Das neue Softwareinformationssystem ist als neues Modul in die Software zedas®asset der Firma PC-Soft GmbH zu integrieren.

Der Entwurfsansatz behandelt folgende Schwerpunkte:

- die Ermittlung der wesentlichen Abläufe, die bei der Anlagendiagnose mit beispielsweise der Hauptkomponentenanalyse durch den Nutzer vollzogen werden
- Beachtung der unterschiedlichen Einsatzszenarien der Software (Einsatz der Software auf einer Bord-Unit mit begrenzten Ressourcen oder serverbasierter Einsatz mit relativ komfortablen Ressourcen)
- das Aufzeigen einer zukunftsorientierten Softwarearchitektur, welche den Anforderungen an Erweiterbarkeit, Modularität standhält und mit der sich moderne Konzepte des Softwaredesigns umsetzen lassen
- die Nutzung wiederverwendbarer, erweiterbarer Softwarekomponenten, wie Patterns und Frameworks.

Folgende Ziele sollen u.a. mit dem Architekturentwurf erreicht werden:

- **Sicherheit**
Es gibt im Wesentlichen zwei Nutzergruppen - die Operativingenieure für das operative Bedienen und die Systemingenieure für tiefer gehende Analysen sowie für die Parametrierung und Konfiguration des Systems. Die Rechte und Rollen müssen beiden Gruppen separat zuweisbar sein. Jeder Nutzer darf nur die Funktionen starten, für die er die Rechte hat.
- **Performanz**
Insgesamt soll die Bedienung des Systems in flüssiger Form und die Verarbeitung und Auswertung der Daten möglichst zeitnah (nach deren Eintreffen) erfolgen. Im Regelfall ist bei täglicher mehrmaliger Verarbeitung/Eintreffen der Datenvektoren nicht mit hohen Prozessorlasten zu rechnen und somit auch eine gute Gesamtperformanz vorhanden. Erhöhte Last kann jedoch auftreten, wenn das Datenaufkommen enorm ansteigt. Das kommt z.B. vor, wenn die zu überwachenden Daten nur einmal täglich (dann jedoch in großer Menge) zur Auswertung bereitgestellt werden. Beim Anlernen des Systems ist ebenfalls mit einer erhöhten Prozessorlast zu rechnen. Diese zuletzt genannten Punkte können die Gesamtleistung des Systems kurzfristig beeinträchtigen.
- **Verfügbarkeit**
Das System muss für den Fall der on-line-Überwachung stets verfügbar sein, d.h. für die Verarbeitung der Daten bereitstehen. Bei hohem Datenaufkommen bzw. sehr kurzen Verarbeitungsintervallen kann es u.U. zu kurzen Latenzen bei der weiteren Abarbeitung kommen.
- **Bedienung**
Diese erfolgt durch den Nutzer über ein browserbasiertes GUI. Die derzeit grundlegenden bekannten Bestandteile sind je ein GUI für das Trainingsmodul und für die on-line-Überwachung/ Datenauswertung. Anhand der definierten Use-Cases wird das GUI für das Anlernmodul einen Wizard-ähnlichen Charakter haben.
- **Testbarkeit**
Aufgrund der webbasierten Technologie gibt es eine Vielzahl zugeschnittener Testumgebungen. Für funktionale, Use-Case getriebene Tests wird „Selenium“ empfohlen (auch aufgrund des guten Firefox-Plugins), für Last- und Performance-Tests sei „JMeter“ genannt. Die hier aufgeführten Testframeworks sind frei verfügbar.
- **Überwachung** In der Software soll ein Monitoring des Zustandes des Softwaresystems enthalten sein. Konkrete zu überwachende Parameter des Systems sind noch nicht ausformuliert, sollen aber über eine REST-API zugänglich und in einem gewissen Umfang auch steuerbar gemacht werden.
- **Robustheit**
Die Software soll einen wartungsfreien Betrieb garantieren und stabil gegenüber unerwartetem Herunterfahren des Systems sein.

- Entkopplung
Die Verarbeitung der Daten darf nicht vom Zustande der Datenquelle abhängen.

5.2 Funktionelle Abläufe

In diesem Abschnitt wird die Funktionalität der Software beschrieben und mit Ablaufdiagrammen in Top-Down-Struktur veranschaulicht. Der erste Teil dient der Bestimmung von Parametern, die später für die Überwachung benötigt werden. Dieser auch als Anlernen (off-line Training) bezeichnete Teil ist in der Regel ein off-line-Prozess. Im Gegensatz dazu erfolgt die Überwachung on-line. In der Abbildung A.1 sind die Teilprozesse dargestellt.

Nutzerinterview

Durch das Nutzerinterview sollen Informationen gesammelt werden, die aus den angelernten Daten nicht hervorgehen, aber für die Bestimmung der zu lösenden Problemstellung des Prozessverhaltens und der geeigneten Diagnosemethode notwendig sind.

Charakterisierung des Prozesses

Bei der Charakterisierung des Prozesses werden über eine Abfrage die wesentlichen Eigenschaften des zu überwachenden Prozesses ermittelt. Diese Informationen dienen u.a. zur Generierung prozessspezifischer Kennwerte. Die Abbildung A.2 stellt den Ablauf schematisch dar.

5.3 Nutzung von Frameworks

Wie bereits beschrieben sollen durch die Nutzung von Frameworks vor allem Zeit und Kosten gespart sowie die Fehleranfälligkeit reduziert werden. Nachfolgend sind Beispiele grundlegender Frameworks aufgeführt.

Auswahl des Container-Frameworks

Als Container-Framework für Servlets können zu Einsatz kommen:

- Tomcat bei leichtgewichtigen Frameworks, wie JSF und Wicket; insbesondere für den Einsatz auf einer Board-Unit;
- Webcontainer im JBoss bei Einsatz parallel zu zedas®asset auf einem Server

Auswahl des Numerik-Frameworks

Das zu einzusetzende Numerik-Framework muss sich ideal in die vorgeschlagene Architektur integrieren lassen sowie die wesentlichen mathematischen und statistischen Operationen unterstützen. Da für die Umsetzung der Software eine verteilte Architektur auf der Basis von Java vorgeschlagen wurde, ist es nur logisch, dass das Numerik-Framework ebenfalls auf der Basis von Java entwickelt sein sollte.

Des Weiteren muss es folgende grundlegende mathematische und statistische Operationen und Methoden unterstützen:

- Basis-Matrizenoperationen, insbesondere Matrizenmultiplikation
- Korrelations- und Kovarianzmatrix
- Standardabweichung
- Eigenwertzerlegung bzw. Singular Value Decomposition (SVD).

In [50] wurden die verfügbaren numerischen bzw. statistischen Frameworks recherchiert und anschließend die Umsetzbarkeit der spezifizierten Statistik-Verfahren mit den recherchierten Frameworks überprüft. Als Hauptquelle dieser Recherche diente der Bereich Java Numerics [18] der Mathematical and Computational Sciences Division (MCSD) des National Institute of Standards and Technology (NIST). Getestet wurden im Rahmen einer Referenzimplementierung nicht nur die Laufzeiten bei den verschiedenen mathematischen- und statistischen Operationen, sondern auch die Dokumentation und Weiterentwicklung der Frameworks. Besonders gute Ergebnisse erreichte im Rahmen dieser Bewertung das Colt-Framework. Colt ist eine Sammlung von wissenschaftlichen und technischen Open Source Java-Bibliotheken, welche auf hohe Performance ausgelegt sind. In ihr sind zum Teil bereits grundlegende Operationen parallelisiert und im Parallel-Colt-Framework zusammengefasst. Alle Packages des Colt Frameworks sind unter dem Copyright des CERN veröffentlicht, aber frei verfügbar. Deshalb wird empfohlen, dieses Framework für die Entwicklung des Softwareinformationssystems zu verwenden.

5.4 Architekturentwurf

Auf Grundlage der in Abschnitt 3.4.3.2 beschriebenen klassischen Multi tier Architektur wurde ein Architekturentwurf für das zukünftige zedas®asset Diagnosemodul entwickelt. Die Abbildung 5.2 stellt die benötigten Komponenten dieses Entwurfes dar. Von großer Bedeutung ist der für die Middle tier vorgesehene Servlet-Container

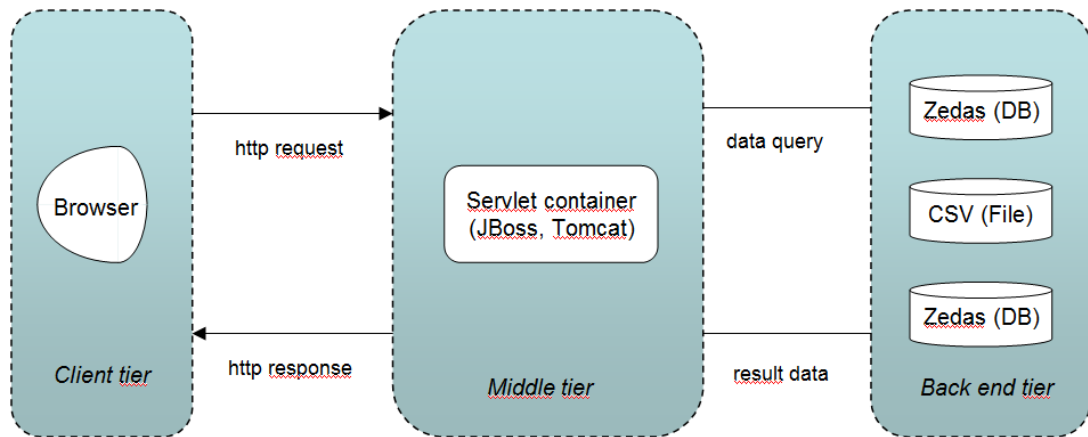


Abb. 5.2: Multi tier Architektur

In einer detaillierteren Sicht ist die Middle tier in zwei weitere Schichten aufgeteilt, in einen webspezifische und einen domänenspezifischen Teil bzw. in eine Web- und eine Business logic tier, dargestellt in der Abbildung 5.3.

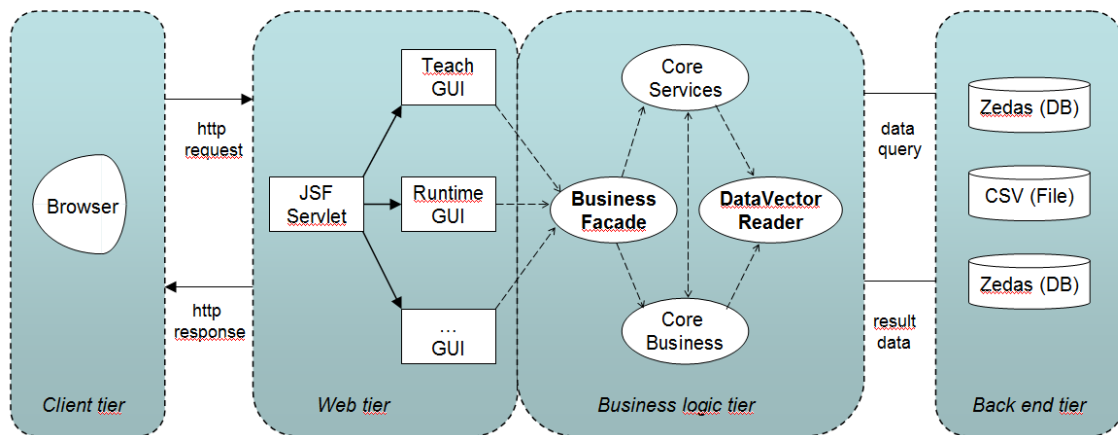


Abb. 5.3: Aufteilung der Middle tier in Web tier und Business tier

Die nachfolgenden Abschnitte beschreiben die einzelnen Schichten.

5.4.1 Client tier

Als Client für die Applikation ist primär ein Web-Browser vorgesehen.

Mit ihm verbinden sich folgende Vorteile:

- Der Nutzer kann mit gewohnter (teilweise personalisierter) Software arbeiten.

- Er bietet eine Reihe bereits vorhandener hilfreicher Features, wie z.B. Suchen, Bookmarks u.s.w..
- Er besitzt bereits vordefinierte Lokalisierungseinstellungen (Browser-Locale), die nicht extra programmiert werden müssen.

Trotz der benannten Vorteile sollte der Einsatz von alternativen Desktop-GUIs ebenfalls möglich sein. Insbesondere bei komplexen Chart-Darstellungen und Bearbeitungsoptionen ist ein Desktop-GUI denkbar. Um diese beiden Client-Varianten zu ermöglichen, wurden das Design und die Geschäftslogik der nachfolgenden Middle tier entsprechend angepasst.

5.4.2 Middle tier

Die mittlere Schicht enthält einerseits die Schnittstelle zur Client Schicht, andererseits auch die Geschäftslogik. Deswegen wurde sie, wie in der Abbildung 5.5 ersichtlich, in eine Web tier und eine Business logic tier unterteilt. Entsprechend den Anforderungen aus Kapitel 3.4.2 soll die Software sowohl auf einem Zedas-Server-System als auch auf einer On-Board-Unit die einlaufenden Prozessdaten mit multivariaten Methoden auswerten und im Falle des Auftretens von Abweichungen vom Normalverhalten entsprechende Meldungen automatisch ausgeben. Aus Gründen der Wirtschaftlichkeit wurde die Architektur so gewählt, dass die benötigten Softwarekomponenten nur einmal entwickelt werden müssen, aber in beiden Varianten eingesetzt werden können. Deshalb soll die Middle tier, wie in Abbildung 5.4 dargestellt, einen Servlet-Container (Servlet 2.5/ JSP 2.1 konform) enthalten. Das heißt, die Frameworks Tomcat ab Version 6 oder JBoss ab Version 5 können hier zum Einsatz kommen.

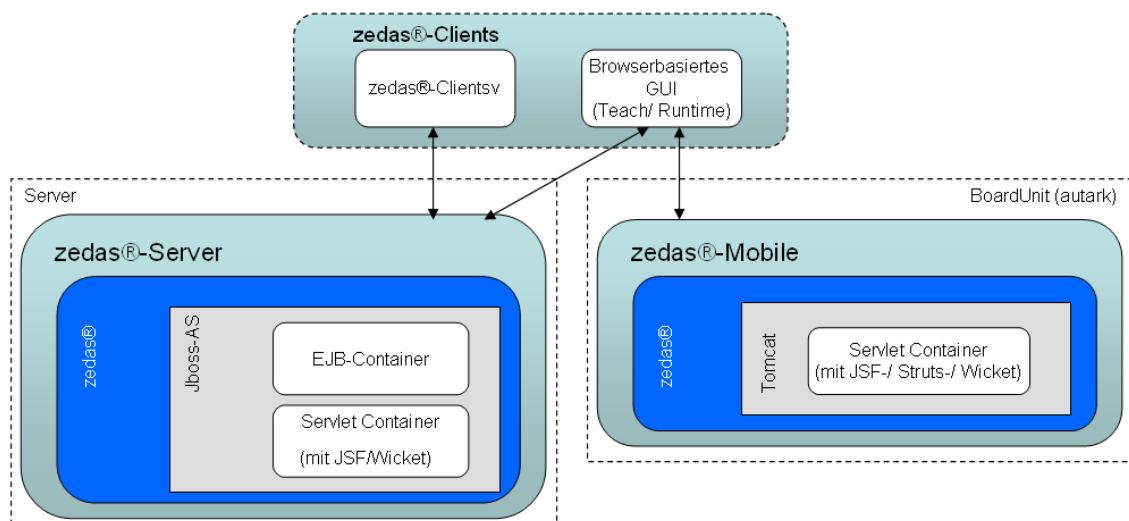


Abb. 5.4: Einsatzszenarien für die Middle tier

Somit besteht die Möglichkeit die Software sowohl auf einem Zedas-System (mit JBoss) als auch auf einer On-Board-Unit (mit leichtgewichtigen Tomcat) zu instal-

lieren. Der Tomcat fungiert in diesem Fall als reiner Servlet-Container. JBoss dagegen besitzt zusätzlich noch einen EJB-Container mit einem hauseigenen, auf Tomcat 6 basierenden, Servlet-Container namens JBossWeb. Die in Abbildung 5.3 gezeigten „CoreServices“ beinhalten vorerst nur den „MathService“, können aber später durch weitere grundlegende Services ergänzt werden. „CoreBusiness“ ist zunächst durch „Analysis“, „Processor Result Actions“, „Runtime-“ und „Teach Modul“ repräsentiert. Gesondert zu betrachten sind „BusinessFacade“ und „DataVectorReader“.

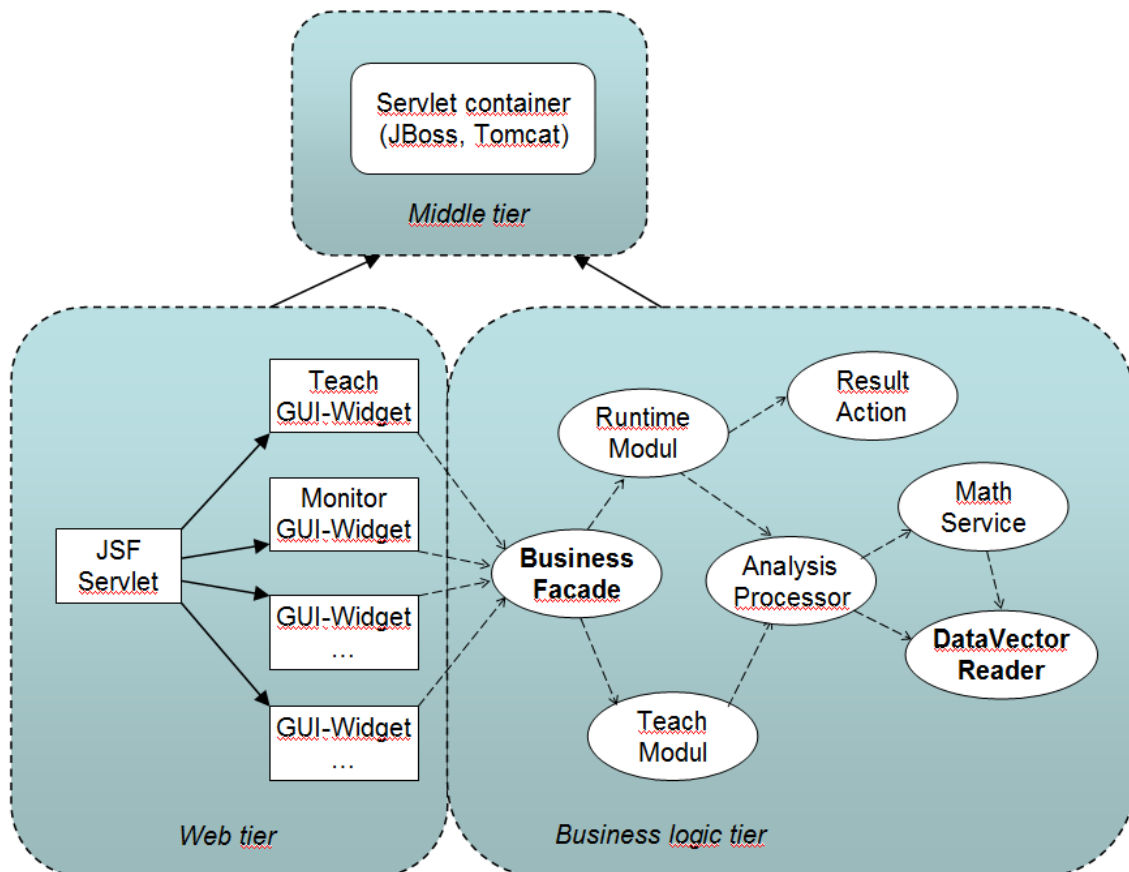


Abb. 5.5: Aufteilung der Middle tier in Web tier und Business logic tier

Nachfolgend werden Web tier und Business logic tier näher beschrieben.

Web tier

Die Web-Schicht ist notwendig, wenn ein Webbrowser als Client verwendet wird. Gerade in dem Bereich browserbasierter GUIs gibt es eine Vielzahl neuer Frameworks, welche Unterstützung bei der Entwicklung von browserbasierten Oberflächen bieten und dem Entwickler viel Arbeit abnehmen. Stellvertretend seien hier die JSF (Java Server Faces), dokumentiert in [12], erwähnt, die bereits in der Version JSF 2.0 vorliegen. JSF als modernes MVC-Framework basierend auf Java Server Pages bringt viele Konzepte mit, welche sich bereits in der Desktop-GUI-Entwicklung bewährt haben,

wie GUI-Elemente, ActionHandler und Listener-Konzept. JSF ist bereits Bestandteil des JBoss. Sollte die Software auf einer kleinen On-Board-Unit installiert werden, kommt nicht der riesige JBoss-Application-Server zum Einsatz, sondern ein schmaler Tomcat. In diesem Fall muss JSF manuell nachgerüstet werden. Tomcat, ein Produkt der Apache Software Foundation, stellt einen Servlet-Container zur Verfügung, mit dessen Hilfe Java-Code auf Webservern ausgeführt werden kann.

Die in der Abbildung dargestellte Webschicht ist vereinfacht. Jedes einzelne GUI-Widget setzt sich, bei konsequenter und sauberer Umsetzung aus drei Hauptkomponenten zusammen, die sich aus dem MVC-Konzept, vorgestellt in Abschnitt 3.4.3.2, ergeben.

Diese sind:

Model

Das Model stellt die Daten für das GUI-Element bereit. Diese sollten einfache DTOs (Data Transfer Objects) sein oder zumindest Objekte, die ausschließlich Parameter und wenig „Intelligenz“ enthalten.

View

Die View dient der Präsentation bzw. Darstellung der Daten. JSF in Version 2.0 setzt hier eindeutig auf XHTML, ein stark an HTML angelehntes Format, welches jedoch „wohlgeformt“ sein muss, um die XML-Verarbeitung zu gewährleisten.

Controller

Der Controller ist u.a. verantwortlich für das Aufbereiten des Models oder Empfangen und Verarbeiten von Events. JSF sieht hierfür das Konstrukt der „ManagedBean“ vor (Objekte, die speziell für diese Aufgabe entwickelt wurden und über die dafür notwendige „Intelligenz“ verfügen, wie implizite Instanziierung, Lebenszyklus, ...).

Business logic tier

Diese Schicht enthält die Geschäftslogik und sollte keinerlei Abhängigkeiten zu Web- oder Back end tier aufweisen. Bei korrekter und sauberer Umsetzung ist es möglich, die Schicht der Geschäftslogik komplett zu entnehmen und in ein anderes Produkt (mit anderer GUI wie z.B. Desktop) zu integrieren.

Business facade

Eine wichtige Komponente der Geschäftslogik (Business logic) ist die Business facade. Sie isoliert den Client von Änderungen in Business Objekten. Bei Weiterentwicklungen der Software stellt sie für die Business logic tier eine stabile Sicht nach „außen“ dar.

Nachfolgend sind die wichtigsten Eigenschaften der Business facade aufgelistet:

- Sie ist Bestandteil des Business layers oder Web layers.
- Sie kapselt webspezifischer Objekte von Businessobjekten.
- Die Business facade definiert Kommunikationsmethoden zwischen beiden Schichten.
- Sie kommuniziert mittels DTOs (DTO - Data transfer object).
- Sie vereinfacht die Handhabung des Business layers nach außen.
- Sie stellt ein bewährtes Design-Pattern dar.

DataVectorReader

Die Komponente DataVectorReader sorgt für eine enge Anbindung der Business logic tier an die Back end tier. Den Vorschlag für das dazugehörige Klassendesign kann man der Abbildung [B.1](#) im Anhang entnehmen. Sie besitzt nachfolgend aufgeführte wesentliche Eigenschaften:

- Sie dient als einheitliche Schnittstelle (Interface) für alle Arten von Datenlesen und Datenvorverarbeitung.
- Sie kapselt die Zugriffe auf Datenquellen/Datensenken.
- Es sind beliebig viele DataVectorReader „in Reihe“ kombinierbar.
- Sie ermöglicht eine gepufferte oder ungepufferte Vorverarbeitung der Daten.
- Das Design wurde angelehnt an das bewährte java.io.Reader Konzept (Decorator pattern).

MathService

Der MathService kapselt die grundlegenden mathematischen Algorithmen bzw. die dafür notwendigen Ersteller der Konfigurationen (siehe Abbildung B.2 im Anhang). Die enthaltenen Services und Factories sind entsprechend dem Singleton-Pattern designet, welches verhindert, dass von einer Klasse mehr als ein Objekt erzeugt wird. Builder-Klassen werden von Services und Factories genutzt und bilden einzelne Teilstrategien ab, die sehr gut austauschbar oder ergänzbar sind. Soll z.B. der Algorithmus für das Anlernen eines PCA-Modells ausgetauscht werden, wird nicht die Factory oder der Service ausgetauscht, sondern lediglich der Builder. Mehrere Builder können für eine Strategie verwendet werden. Der Nutzer kann auf diese Art entscheiden, welchen konkreten Algorithmus er nutzen möchte.

5.4.3 Back end tier

Die Backend-Schicht birgt eine Reihe von Datenquellen und -senken. Sie muss Daten aus einem Zedas-Datenbank-System, aber auch aus CSV-Dateien bzw. zukünftig auch von OPC-Datenquellen lesen und weiterverarbeiten können. Aufgrund dieser Vielfältigkeit ist eine klare Abgrenzung der Business logic tier zur Back end tier nicht immer eindeutig. Insbesondere bei CSV-Dateien verschwimmen die Grenzen, da die Business-Schicht in diesem Fall den direkten Zugriff auf CSV-Dateien selbst implementiert. Im Gegensatz dazu ist ein Zedas-Datenbank-System klar abgetrennt, da hier der EJB-Container (EJB - Enterprise Java Bean) bzw. Hibernate-Komponenten (Open-Source Persistenz-Framework für das Object-relationale Mapping dokumentiert in [40]) auf die Datenbank zugreifen. Die Backend-Teile können beispielsweise als Fat-Client (Java-Library) ausgelegt sein. Die Struktur lässt ebenso die Auslegung als verteiltes System zu.

5.5 Zusammenfassung

Der Einsatz einer verteilten Architektur mit browserbasierten Clients bringt u.a. folgende Vorteile mit sich:

- Er ermöglicht den Einsatz von REST-Clients/Schnittstellen (REST - Representational State Transfer).
- Er sichert dem Nutzer Komfortmerkmale browserbasierter GUIs, wie Bookmarking seitenbasierte Suchmasken, Metainformationen,
- Die großen Prozessdatenmengen „Massendaten“ bewegen sich nur zwischen den Connectoren und dem Backend.
- Die Steuerung von Aktionen erfolgt vom GUI aus.
- Die Numerik-Engine ist austauschbar.

Insgesamt besitzt dieser Architekturvorschlag eine gute Zukunftsperspektive, da moderne Ansätze, bewährte Standard-Design-Patterns und etablierte Frameworks verwendet werden. Die einzelnen Komponenten sind durch Kapselung austausch- bzw. erweiterbar.

Der Designvorschlag ist im Rahmen der weiteren Entwicklung zu vertiefen. Für die vorgeschlagenen Softwarekomponenten muss das konkrete Klassendesign entworfen und mit Ablauf- und Sequenzdiagrammen untersetzt werden. Ein weiterer entscheidender Schritt ist der Entwurf der webbasierten Nutzeroberfläche. Die funktionellen Abläufe bieten dafür eine gute Grundlage. Da das den Rahmen dieser Arbeit deutlich überschreitet, wurden nur die wichtigen grundlegenden Ansätze aufgeführt und in ausgewählten Einzelfällen, wie z.B. dem `DataVectorReader` aus der Business logic tier, bis zum Klassendesign im Anhang dargestellt.

6 Anwendung der Verfahren an einer Gurtförderanlage

Die in den Kapiteln 3 und 4 vorgestellten multivariaten statistischen Fehlerdetektionsverfahren wurden am Beispiel einer Bandanlage im chilenischen Kupferbergbau getestet. Anhand realer Messdaten eines Kupfererz-Transportprozesses sollen die unterschiedlichen Methoden der Datenaufbereitung und anschließender Fehlerdetektion unter industriellen Bedingungen angewendet werden. Die Messdaten wurden sowohl während des fehlerfreien als auch während des fehlerbehafteten Betriebes der Gurtförderanlage aufgenommen. Zum Schluss erfolgt eine kurze Diskussion zu den mit der Implementierung der Methoden zusammenhängenden Problemen und Chancen.

6.1 Beschreibung der Gurtförderanlage

Das untersuchte Massentransportsystem ist Bestandteil eines Leach-Pad Systems und dient zum Auf- und Abhalten von Kupferoxid. Dabei handelt es sich um einen stationären Langstreckenförderer. Die Antriebsstation besitzt zwei Antriebstrommeln, wovon eine mit zwei Antriebsmotoren, die andere mit nur einem Antriebsmotor angetrieben wird. Der Transport des Schüttgutes erfolgt über einen Fördergurt mit einer Breite von 1524 mm (60") und über eine Distanz von 1600 m. Das Fördergut wird mit einer Geschwindigkeit von 4 m/s transportiert. Die nachfolgende Abbildung verdeutlicht den schematischen Aufbau der Gurtförderanlage.

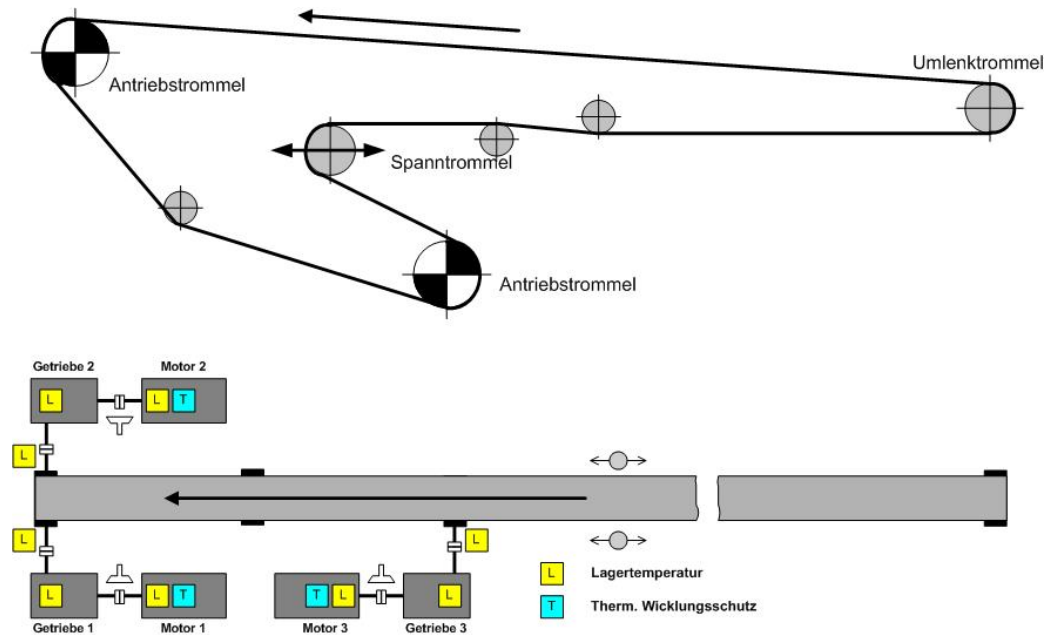


Abb. 6.1: Schematischer Aufbau der Bandantriebsstation

Hier noch mal eine Zusammenstellung der wichtigsten Daten zur untersuchten Gurtförderanlage:

Ort der Anlage:	Bergwerk Chile
Art der Anlage:	stationärer Langstreckenförderer
Bandbreite:	1524 mm (60")
Bandgeschwindigkeit:	4 m/s
Abstand Trommel/Trommel:	1930 m
Antriebstrommel DM mit Belag:	1040 mm
Umlenktrommel DM mit Belag:	826 mm
Antriebsleistung:	1500 kW (3x500 kW)
Motor nach NEMA:	3-Phasen Induktion
Schüttgut:	Kupferoxid
Transportleistung:	4200 t/h

Aufbau der Antriebs- und Abwurfstation

- Die Antriebsstation ist auf einer Plattform montiert.
- Die Ausführung erfolgte in torsionssteifer Stahlkonstruktion mit beidseitig angeordnetem Laufsteg, Gitterrostbelag, Fußleisten und Geländer.
- Das Antriebsgerüst besteht aus einer geschweißten, teils geschraubten Profilstahlkonstruktion.

- Die Antriebstrommel besitzt einen Gummireibbelag mit Profil, eine Welle mit wälzgelagerten Stehlagern sowie Spannsatzverbindungen mit dem Trommelboden.
- Die Antriebseinheiten sind auf einer Antriebsschwinge mit Drehmomentenstütze montiert.
- Das Kegelrad-Stirnradgetriebe ist mit einsatzgehärteter und geschliffener Verzahnung in öldichtem Gehäuse mit Wälzlagerung und Tauchschmierung sowie mit einem Lüfter auf der Eingangswelle ausgerüstet.
- Das Getriebe ist über eine starre Kupplung mit der Trommelwelle verbunden.
- Die Kraftübertragung zwischen Motor und Getriebe erfolgt über eine Strömungskupplung mit obligatorischer Schmelzschraube und zusätzlicher berührungsloser thermischer Überwachung.
- Zur Begrenzung der Nachlaufwege ist in die Verbindung zwischen Motor und Getriebe eine Bremse integriert.
- Die elektrohydraulische Bremse ist mit einer Verschleißüberwachung ausgerüstet.

Aufbau der Umlenkstation mit Aufgabetrichter

- Das Umlenkgerüst ist in geschweißter, teils geschraubter Profilstahlkonstruktion ausgeführt und auf ein Fundament montiert.
- Die Umlenktrommel hat einen Gummibelag, eine Welle mit wälzgelagerten Stehlagern sowie Spannsatzverbindungen mit dem Trommelboden.
- Die Innengurtreinigerstation ist mit pflugförmig ausgebildeten Reinigungseinrichtungen ausgerüstet.
- Der Aufgabetrichter besteht aus einer geschweißten, teils geschraubten Profilstahlkonstruktion, die auf ein Fundament montiert wurde.
- Die Aufgabeeinheit ist mit Gummiabdichtungen und Gummivorhängen sowie mit Schweißblechen ausgekleidet.



Abb. 6.2: Gurtförderanlage Chile Kupferbergbau

Die technischen Daten und Anlagenparameter der Gurtförderanlage CV01 beruhen auf den bereitgestellten Datenblättern und Angaben des Anlagenherstellers.

6.2 Untersucher Produktionsdatensatz

Für die Untersuchung standen Messwerte aus einem realen Produktionsprozess über den Zeitraum eines halben Jahres, von Dezember 2008 bis Mai 2009 zur Verfügung sowie eine Liste der in diesem Zeitraum aufgetretenen Anlagenstörungen. Von den 57 Variablen wurden diejenigen verworfen, die innerhalb einer Zeitspanne zu wenige Werte aufwiesen, sodass insgesamt nur noch 23 Datenspuren für die Überwachung verwendet werden konnten. Wie man der Übersicht in Tabelle 6.1 entnehmen kann, sind in den Daten Beobachtungen verschiedenster Vibrations- und Temperatur sowie Motorstromsensoren enthalten. Leider konnte der Anlagenbetreiber keine Strom-, Leistungs- und Temperaturdaten der Antriebsmotoren 2 und 3 zur Verfügung stellen.

Die Abstraten der einzelnen Datenvektoren sind durch den bei der Langzeitspeicherung verwendeten Komprimierungsalgorithmus (Speicherung nur bei Wertänderung über 1%) unterschiedlich groß. Des Weiteren enthalten die Daten sowohl stationäre als auch instationäre Zustände, wie An- und Abfahrvorgänge.

Drive pulley 1 bearing T left TE1301-R01	°C
Drive pulley 1 bearing T right TE1302-R02	°C
Drive pulley 2 bearing T left TE1303-R01	°C
Drive pulley 2 bearing T right TE1304-R02	°C
Motor 1 gearbox temperature TE1321-R01	°C
Representative bearing T Motor 1	°C
Representative temperature drive pulley 1	°C
Representative winding T Motor 1	°C
Motor 1 Average Phase Current	A
Motor 1 Phase A Current	A
Motor 1 Phase B Current	A
Motor 1 Phase C Current	A
Motor 1 Real Power in kW	kW
Drive pulley 1 representative vibration	m/s ²
Drive pulley 1 vibration left VE1401-B01	m/s ²
Drive pulley 1 vibration right VE1402-B01	m/s ²
Drive pulley 2 representative vibration	m/s ²
Drive pulley 2 vibration left VE1403-B01	m/s ²
Representative vibration gearbox motor 1	m/s ²
Vibration gearbox 1 Sensor 1 VE1421-B01	m/s ²
Vibration gearbox 1 Sensor 2 VE1422-B01	m/s ²
Vibration motor 1 VE1411-B01	m/s ²
Vibration motor 1 VE1430-B01	m/s ²

Tabelle 6.1: Liste verwendeter Variablen der Gurtbandförderanlage CV01 nach Angaben des Anlagenherstellers

Die Daten wurden deshalb mit folgenden in den Kapiteln 3.2 und 4.2.2 vorgestellten Verfahren zur Datenvorverarbeitung aufbereitet:

- Datendekomprimierung durch Interpolation und Extrapolation an den Rändern
- Datenimputation
- Eliminierung der Ausreißer
- Herausschneiden der An- und Abfahrvorgänge.

Die nachfolgenden Abbildungen zeigen ausgewählte Beobachtungen vor und nach der Datenvorverarbeitung.

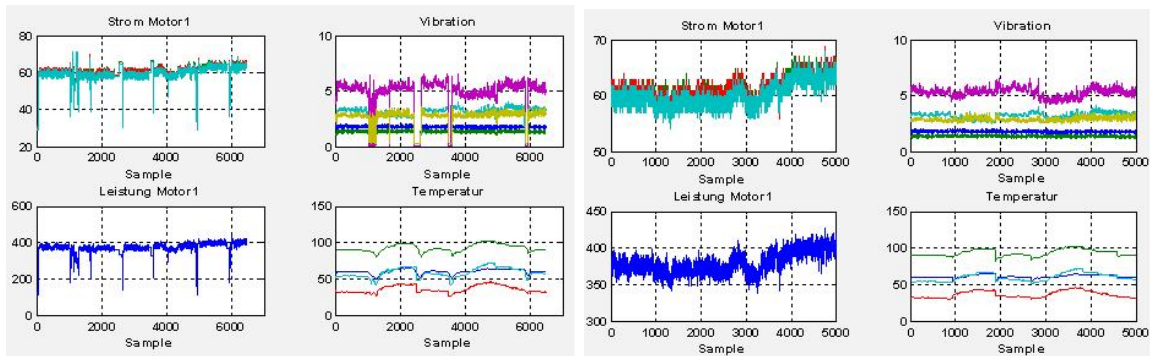


Abb. 6.3: Ausgewählte Variablen ohne Datenvorverarbeitung

Abb. 6.4: Ausgewählte Variablen nach der Datenvorverarbeitung

6.3 Fehlerzustände

Für die Bewertung der Leistungsfähigkeit der entwickelten Verfahren wurden Daten realer Fehlerzustände der Gurtförderanlage CV01 analysiert.

Folgende Fehlerbilder standen zur Verfügung:

- Ausfall der Antriebstrommel 1 durch Wellenbruch
- Mantelriss an der Antriebstrommel 1
- Sensorfehler
- erhöhte Schwingungen am Antriebssystem

Die Auswirkungen auf das tatsächliche Betriebsverhalten der Gurtförderanlage sind bezogen auf die einzelnen Fehler recht unterschiedlich. Während der Wellenbruch und der Mantelriss an der Antriebstrommel 1 nach ihrer Entdeckung zum Abschalten und zur anschließenden Reparatur der Anlage führten, wurde bei den anderen beiden Fehlern der Anlagenbetrieb bis zur planmäßigen Wartung fortgesetzt. Die nächsten Abschnitte enthalten kurze Beschreibungen der genannten Fehlerbilder.

6.3.1 Ausfall der Antriebstrommel 1 durch Wellenbruch

Ein Wellenbruch an der Antriebsstation einer Gurtförderanlage ist ein sehr schwerwiegendes aber auch seltenes Ereignis, welches nicht sofort zum Totalausfall des Gesamtsystems führen muss. Durch die Ausstattung der Antriebsstation mit zwei Antriebstrommeln, einem linken und rechten Motor an der Antriebstrommel 1 und einem Motor an der Antriebstrommel 2 ist die Gurtförderanlage durchaus in der Lage, ihre eigentliche Aufgabe, den Transport von Schüttgütern, weiter auszuführen. Das kann zum Beispiel vorkommen, wenn der Fehlerzustand infolge unzureichender Mittel zur Fehlerdetektion nicht bemerkt wird oder der Betrieb eine Zeit lang fortgesetzt werden muss, wie zum Entladen der Anlage. Infolge der fehlerbedingten Unwuchten

und Schwingungen werden aber eine Reihe weiterer Komponenten (Lager, Motoren, ...) nachhaltig geschädigt.

Der Wellenbruch an der Gurtbandförderer CV-01 wurde in der Nacht vom 07.01. auf den 08.01.2009 bemerkt und infolgedessen die Anlage angehalten. Nach Inspektion der Trommel stellte man einen Bruch der Welle (Wellenzapfen) auf der rechten Seite der Antriebstrommel 1 fest.

6.3.2 Mantelriss

Zu den Hauptursachen für Mantelrisse gehören

- der Verschleiß des Trommelmantels sowie
- äußere Störeinflüsse durch Steine oder Eisenteile.

Wie beim Wellenbruch besteht auch hier die Gefahr der nachhaltigen Schädigung weiterer Komponenten, wie Gurt, Lager und Antriebsmotoren, wenn der Fehler nicht rechtzeitig erkannt wird.

Der Mantelriss an der Antriebstrommel 1 der Anlage CV01 wurde am 06.02.09 durch das Betreiberpersonal bemerkt und die Anlage daraufhin angehalten. Noch am selben Tag erfolgte der Austausch gegen eine neue Antriebstrommel.

6.3.3 Sensorfehler

Wie die in [20] veröffentlichten Statistiken zu Ausfallraten von Anlagenkomponenten eines komplexes mechatronischen Systems zeigen, gehören Sensorfehler mit zu den häufigsten Fehlerursachen. Danach rangiert die Häufigkeit von Ausfällen der Sensoren gleich an dritter Stelle nach den Ausfallhäufigkeiten von elektrischen und regelungstechnischen Komponenten. Die Fehler an Sensoren kann man nach [73] unterscheiden in:

- additiver Fehler
- multiplikativer Fehler
- konstante Werte
- langsame Drift und
- zufälliges Springen der Werte.

Ob der Defekt eines Sensors zum Ausfall des mechatronischen Systems führt, hängt meist davon ab, ob der Sensor in ein Regelungskonzept eingebunden ist. Dann fallen nicht nur die Kosten für den Sensorwechsel selbst, sondern auch meist viel höhere

Kosten durch den Produktionsausfall und die ungeplante Nichtverfügbarkeit der Anlage an. Mittels rechtzeitiger Detektion könnte die Reparatur zum Beispiel in einen Zeitraum planmäßiger Wartung gelegt werden.

An der Gurtbandförderanlage CV01 kam es am 03.03.2009 zu einem Spontanausfall des Temperatursensors am linken Lager der Antriebstrommel 1. Wurden bis zu diesem Zeitpunkt Temperaturen von etwa 25°C bis 50°C gemessen, zeigte das System ab dem Zeitpunkt des Ausfalls konstant null an.

6.3.4 Erhöhte Schwingungen an der Antriebstrommel 2

Die Schwingungsanalyse gehört zu den am häufigsten eingesetzten Condition Monitoring Verfahren. Gemessen werden die durch die Bewegung von Maschinenelementen erzeugten Schwingungen. Bei den Sensoren kann es sich um Beschleunigungs-, Geschwindigkeits- oder Wegmessungssensoren handeln, wobei man die Signalvarianten ineinander umrechnen kann.

Das Bestimmen der Schwingungen an der Anlage CV01 erfolgt durch Körperschallmessung an den aus Sicht der Anlageninstandhaltung sensiblen Punkten. Zu den erfassten Schwingungssignalen gehören:

- Antriebstrommel 1 repräsentative Schwingungen
- Antriebstrommel 1 Schwingungen links
- Antriebstrommel 1 Schwingungen rechts
- Antriebstrommel 2 repräsentative Schwingungen
- Antriebstrommel 2 Schwingungen links
- Antriebstrommel 2 Schwingungen rechts
- Getriebe 1 repräsentative Schwingungen
- Getriebe 1 Sensor 1
- Getriebe 1 Sensor 2
- Schwingungen Motor 1
- Schwingungen Motor 2.

Gemessen und archiviert werden die Beschleunigungswerte der Sensoren. Die Analyse der Frequenzen erfolgt durch den Instandhaltungsdienstleister off-line mithilfe eines separaten Analysegerätes. Das heißt, die Frequenzwerte selbst wurden nicht on-line erfasst und archiviert und standen somit für eine multivariate Fehlererkennung nicht zur Verfügung.

An der Anlage CV01 traten ab dem 12.03.2009 erhöhte Schwingungen an der Antriebstrommel 2 auf. Die genauen Ursachen, welche zu dem Fehlerbild führten, konnten durch den Anlagenbetreiber im Nachhinein nicht mehr ermittelt werden.

6.4 Zusammenfassung

Der im ersten Abschnitt dieses Kapitels beschriebene Aufbau der Gurtförderanlage CV01 kann als typisch für die rohstoffproduzierende Industrie bezeichnet werden. Als Bestandteil eines Leach-Pad Systems ist sie durch die rauen Umgebungsbedingungen, wie Säuredämpfe in der Luft, große Temperaturschwankungen und Staub, extremen Stress und Verschleiß ausgesetzt. Für die Auswertung der Daten standen leider nur eine ausgewählte Anzahl von Sensoren zur Verfügung. Bedingt durch Datenverdichtungsalgorithmen am Ausgang der Prozesssteuerung, kommen die Daten nur sehr unregelmäßig. Das stellt besondere Herausforderungen an die Datenaufbereitung vor der Analyse mit multivariaten Methoden. Trotz dieser Schwierigkeiten muss man festhalten, dass diese Bedingungen sehr oft im realen industriellen Umfeld zu finden sind. Die mit den Prozessdaten aufgenommenen Fehlerzustände betreffen einzelne Komponenten sowie auch die komplette Antriebseinheit. Während ein Wellenbruch und ein Mantelriss an einer Antriebstrommel eher selten zu finden sind, treten Sensorausfälle und erhöhte Schwingungen öfter an mechatronischen Systemen auf. Insofern bilden die Daten eine gute Grundlage, um die Leistungsfähigkeit der vorgestellten und entwickelten Verfahren zu demonstrieren.

7 Ergebnisse

Die in den Kapiteln 3 und 4 vorgestellten multivariaten statistischen Fehlerdetektionsverfahren werden am Beispiel einer Bandanlage im chilenischen Kupferbergbau getestet. Es gilt, die unterschiedlichen Methoden der Datenaufbereitung und Fehlerdetektion anhand realer Messdaten eines Kupfererz-Transportprozesses unter industriellen Bedingungen zu verifizieren. Die Messdaten wurden sowohl während des fehlerfreien als auch während des fehlerbehafteten Betriebes der Gurtförderanlage aufgenommen. Abschließend erfolgt eine kurze Diskussion der mit der Implementierung der Methoden zusammenhängenden Probleme und Chancen.

7.1 Fehlerdetektion mithilfe der PCA

Zunächst werden die im Abschnitt 6 beschriebenen Fehlerzustände mit den in dem Abschnitt 3.3.1 „Fehlererkennung auf Basis der PCA“ beschriebenen Methoden untersucht.

Die Analyse erfolgt über Datensätze mit folgenden Fehlerbildern:

- Ausfall der Antriebstrommel 1 durch Wellenbruch
- Mantelriss an der Antriebstrommel 1
- Sensorfehler
- erhöhte Schwingungen an der Antriebstrommel 2

Alle Datensätze wurden dazu bezüglich des Erwartungswertes und der Varianz normiert.

Zu den Schwerpunkten der Bewertung der Ergebnisse der Fehlerdetektion mittels der PCA gehören

- die sichere Erkennung des Fehlerzustandes mittels der T^2 -Statistik
- die sichere Erkennung des Fehlerzustandes mittels der SPE -Statistik
- die Fehlalarmraten (FAR) im fehlerfreien Zustand sowie
- die Lage der Schwellwerte.

7.1.1 Ausfall der Antriebstrommel 1 durch Wellenbruch

Wie bereits im Abschnitt 6 beschrieben, ist ein Wellenbruch an der Antriebsstation einer Gurtförderanlage ein sehr schwerwiegendes aber auch seltenes Ereignis. Bedingt durch den Aufbau der Antriebsstation mit zwei Antriebstrommeln, je einen linken und einen rechten Motor an der Antriebstrommel 1 sowie einen Motor an der Antriebstrommel 2 war die Gurtförderanlage nicht sofort in ihrem Betrieb gestört, sondern konnte ihren Betrieb eine Zeit lang fortsetzen. Wie die nachfolgenden zwei Abbildungen mit ausgewählten Signalen des Untersuchungszeitraumes zeigen, sind in den Verläufen der Strom und Leistungsdaten der Motoren trotz des Wellenbruchs keine sichtbaren Veränderungen erkennbar. Auch über die gemessenen Schwingungen an der Antriebstrommel 2 kann man nicht sofort auf einen Fehlerzustand schließen. Dagegen zeigt das Vibrationssignal an der Antriebstrommel 1 ab Sample 31000 einen deutlichen Anstieg. Ab Sample 50000 sind die Schwingungen so stark, dass sie auch am Schwingungsmessgeber der Antriebstrommel 2 zu starken Ausschlägen in den Signalamplituden führen.

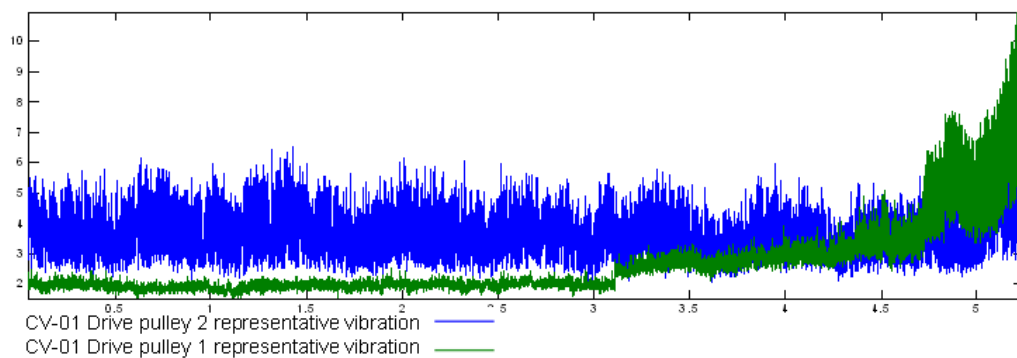


Abb. 7.1: Fehler Wellenbruch an der Antriebstrommel 1: Schwingungen an der Antriebstrommel 1 und 2

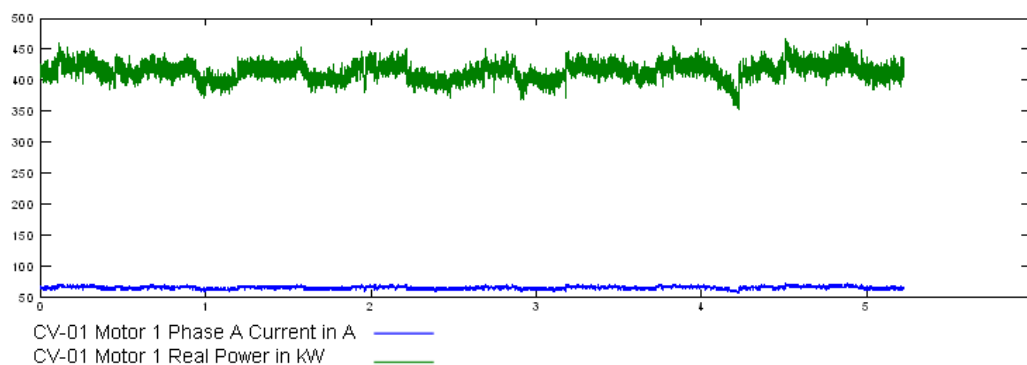


Abb. 7.2: Fehler Wellenbruch an der Antriebstrommel 1: Strom und Leistung Motor 1

Die für die Überwachung mittels der T^2 - und SPE -Statistiken notwendigen statistischen Parameter wurden über das im Abschnitt 3.3.1 beschriebene off-line-Training

von Sample 1 bis 25000 ermittelt. Mittels des Kaiser-Kriteriums bestimmte die Software aus allen Eigenvektoren fünf als Hauptkomponenten. Die folgenden Abbildungen zeigen die T^2 - und SPE -Indizes mit den zugehörigen Schwellwerten. Wegen der großen Unterschiede zwischen den Indexwerten des fehlerfreien und des fehlerbehafteten Bereiches sowie zur genaueren Bewertung des Übergangs zwischen beiden Zuständen wurde eine logarithmische Einteilung der Y-Achse gewählt.

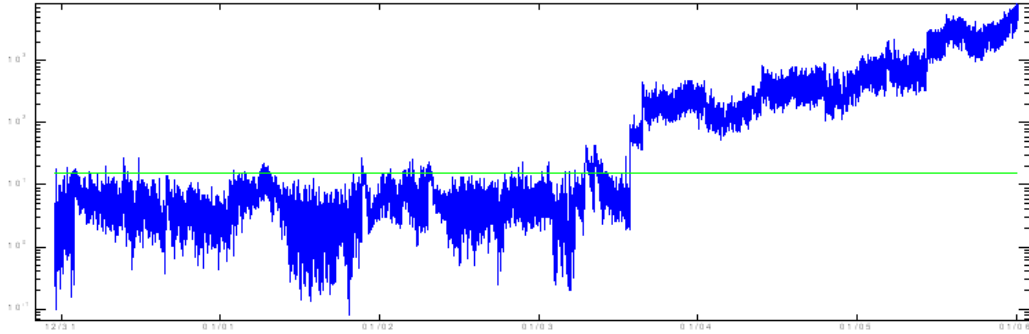


Abb. 7.3: Fehler Wellenbruch an der Antriebstrommel 1: T^2 -Index nach der PCA

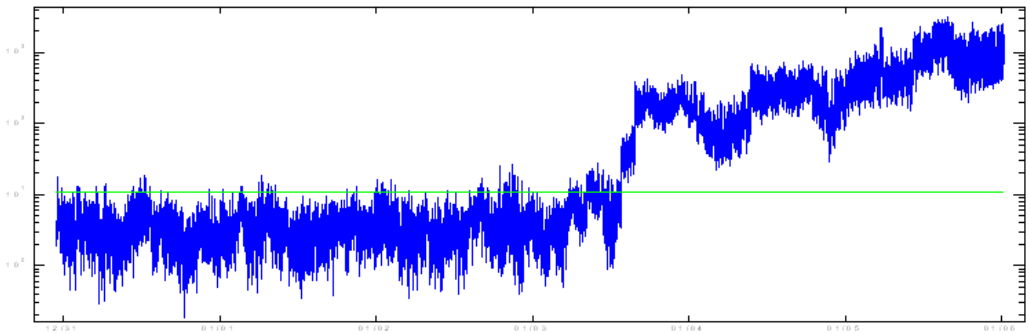


Abb. 7.4: Fehler Wellenbruch an der Antriebstrommel 1: SPE -Index nach der PCA

Wie die Abbildungen 7.3 und 7.4 zeigen, konnte der kritische Betriebszustand der Gurtförderanlage infolge des Wellenbruchs durch das eindeutige Überschreiten der Schwellwerte beider Indizes ab den Abendstunden des 03.01.2009 eindeutig nachgewiesen werden. Da das Bedienpersonal den Fehlerzustand der Anlage erst in der Nacht vom 07.01. zum 08.01.2009 bemerkte und daraufhin die Anlage abschaltete, kann man schlussfolgern, dass eine Fehlerdetektion mittels der PCA bereits etwa vier Tage vorher den kritischen Betriebszustand gemeldet hätte.

Die Abbildungen 7.3 und 7.4 zeigen ebenfalls, dass beide Indizes die Schwellwerte schon im fehlerfreien Zustand (Sample 1-25000) mehrmals überschritten. Auf die möglichen Ursachen wurde bereits im Abschnitt 4.2.2 eingegangen. Die Fehlalarmraten haben für die Sample 1-25000 die Werte $FAR_{T^2} = 1,5\%$ und $FAR_{SPE} = 0,77\%$. Bei einer Abtastrate von etwa 10 Sekunden, würde das System über die T^2 Auswertelogik alle 11 Minuten und über die SPE Auswertelogik alle 21 Minuten einen

Alarm auslösen. Für den praktischen Einsatz bedeutet jeder Fehlalarm unnötigen Aufwand für das Personal, was letztendlich zu einer Verringerung der Akzeptanz eines solchen Überwachungssystems führen kann. So gesehen sind die erzielten Werte zu den Fehlalarmraten viel zu hoch. Da aber die Ausschläge beider Indizes für den fehlerbehafteten Bereich mit Werten von über 10^3 enorm stark sind, könnte man deutlich höhere Schwellwerte wählen, somit Fehlalarme vermeiden und trotzdem den Fehler sicher detektieren.

7.1.2 Mantelriss

Der im Abschnitt 6 beschriebene Mantelriss an der Antriebstrommel 1 der Gurtförderanlage CV01 wurde am 06.02.09 durch das Personal bemerkt. Noch am selben Tag erfolgte der Wechsel der Trommel. Dem Verlauf der Signale von Strom und Leistung des Antriebsmotor 1 und der Schwingungen der Antriebstrommel 1 links und rechts für den Zeitraum vom 02.02. bis zum 05.02.09 in Abbildungen 7.5 und 7.6 kann man entnehmen, dass infolge des Mantelrisses ab Sample 25000 erhöhten Schwingungen an den Vibrationsgebern auf der rechten Seite des Antriebes 1 auftraten. Sowohl bei Strom und Leistung als auch bei den Schwingungsdaten der Antriebstrommel 2 sind keine sichtbaren Veränderungen erkennbar.

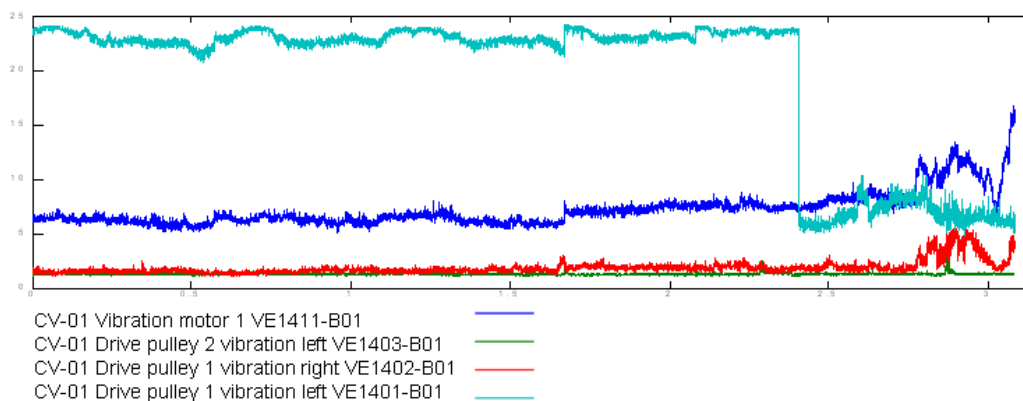


Abb. 7.5: Fehler Mantelriss an der Antriebstrommel 1: Vibrationen

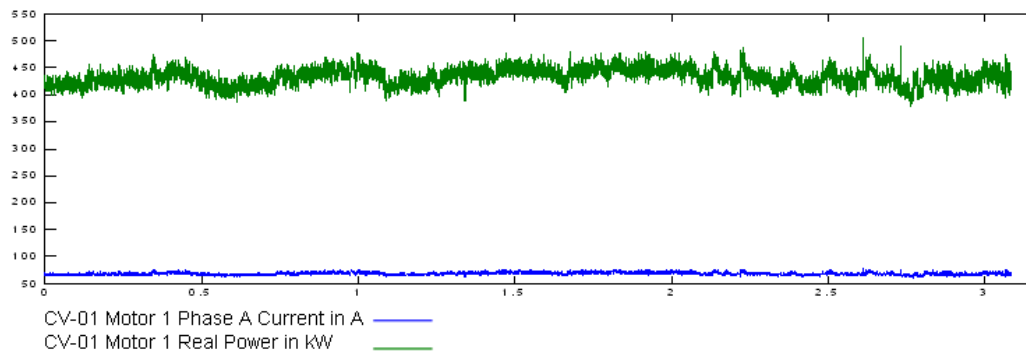


Abb. 7.6: Fehler Mantelriss an der Antriebstrommel 1: Strom und Leistung

Für das off-line-Training wurden die Sample 1 bis 15000 herangezogen und drei Hauptkomponenten bestimmt. Die folgenden Abbildungen enthalten die T^2 - und SPE -Indizes mit den zugehörigen Schwellwerten für den betrachteten Zeitraum.

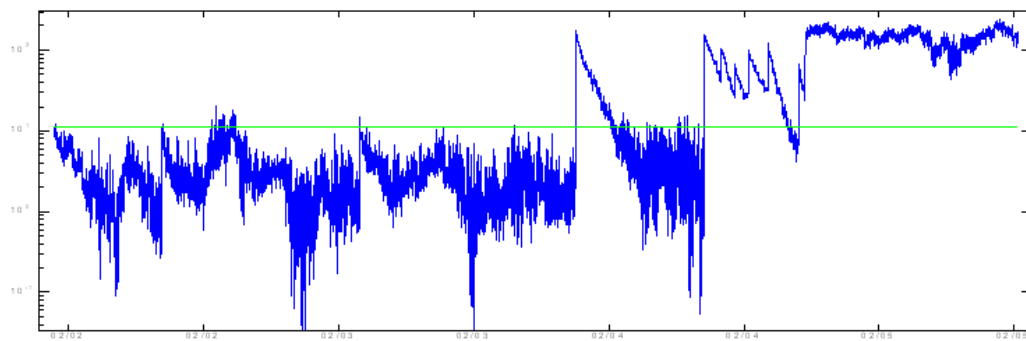


Abb. 7.7: Fehler Mantelriss an der Antriebstrommel 1: T^2 -Index nach PCA

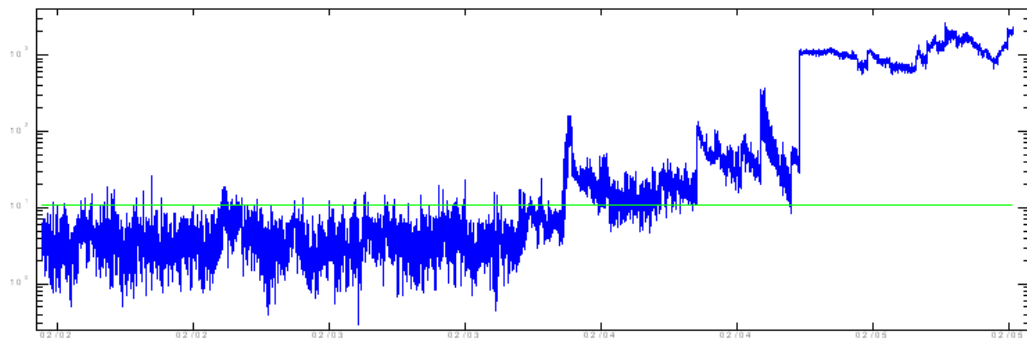


Abb. 7.8: Fehler Mantelriss an der Antriebstrommel 1: SPE -Index nach PCA

In den Verläufen der Indizes in den Abbildungen 7.7 und 7.8 kann man sehen, dass die Veränderung insbesondere der Schwingungsdaten infolge des Mantelrisses zu deutlichen Ausschlägen der Indizes führen. Ähnlich wie beim Wellenbruch haben die Indizes die Schwellwerte bereits zwei Tage vor dem Erkennen des Fehlers durch das Personal dauerhaft überschritten.

Die Fehlalarmraten über die Trainingsdaten betrugen für den T^2 -Index $FAR_{T^2} = 0,9\%$ und für den SPE -Index $FAR_{SPE} = 1,2\%$ und sind damit deutlich zu hoch. Im Fehlerfall überschritten die Amplituden des T^2 -Indexes den Schwellwert um das Zehnfache und den des SPE -Indexes um das Einhundertfache. Folglich könnte man auch hier die Größe der Schwellwerte neu bewerten, um Fehlalarme zu vermeiden.

7.1.3 Sensorfehler

Wie im Abschnitt 6 beschrieben, trat der Spontanausfall des Temperatursensors am linken Lager der Antriebstrommel 1 am 06.03.2009 auf. Der nachfolgenden Abbildung 7.9 kann man entnehmen, dass ab dem Zeitpunkt des Ausfalls am Steuerungssystem der Anlage nur Null-Werte anlagen. Weitere Messgeber waren nicht betroffen. Da der Sensor nicht in das Regelungskonzept der Anlage integriert ist, kam es infolge des Ausfalls auch nicht zu einem Abfahren oder zu eingeschränkter Leistung der Gurtförderanlage.

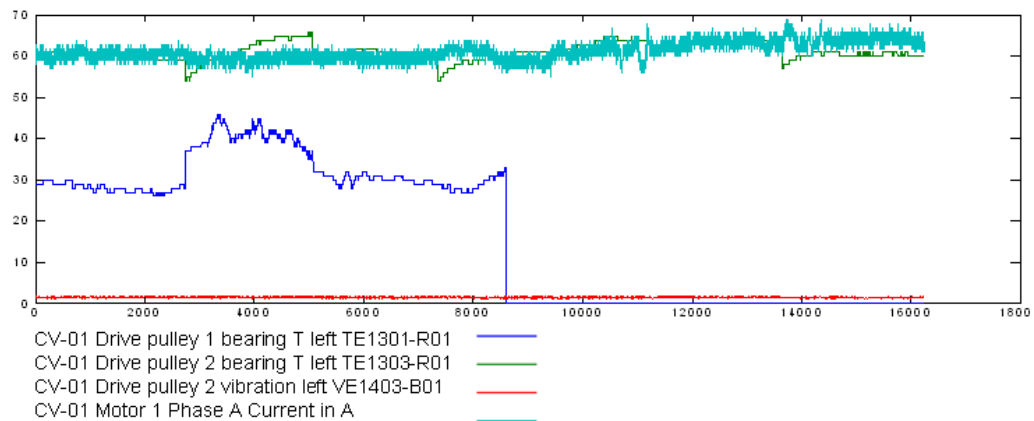


Abb. 7.9: Fehler Sensorausfall: Vibrationsdaten

Das Antrainieren der für die Überwachung notwendigen statistischen Parameter erfolgte mit den Daten von Sample 1 bis 6000. Im Ergebnis wurden vier Hauptkomponenten bestimmt. Die nachfolgenden Abbildungen zeigen die T^2 - und SPE -Indizes mit den zugehörigen Schwellwerten.

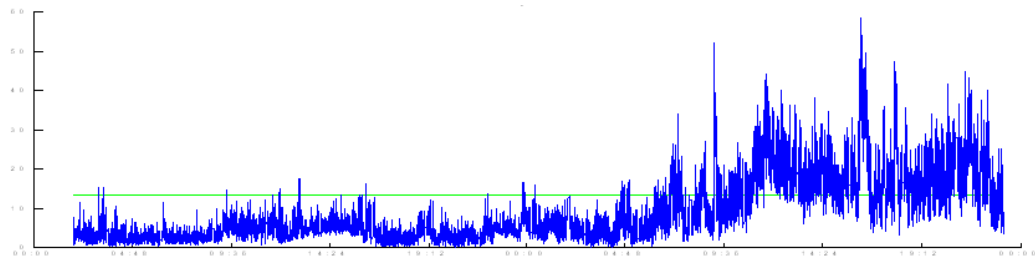


Abb. 7.10: Fehler Sensorausfall: T^2 -Index nach der PCA

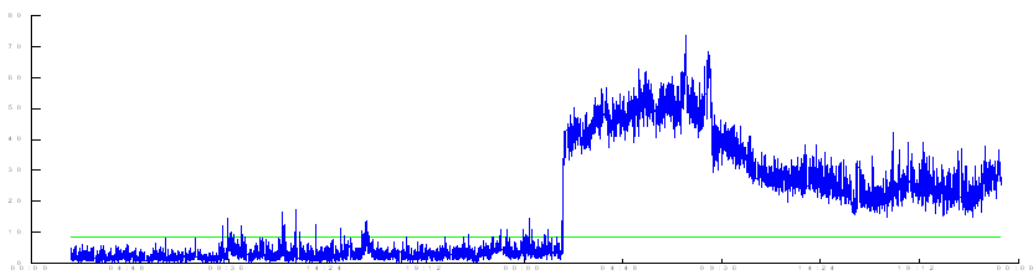


Abb. 7.11: Fehler Sensorausfall: SPE -Index nach der PCA

Der am 03.03.2009 gegen 02.00 Uhr aufgetretene Ausfall des Lagertemperatursensors an der Antriebstrommel 1 konnte nur durch den SPE -Index eindeutig detektiert werden. Die Überschreitungen des Schwellwertes durch den T^2 -Index sind nur geringfügig, nicht dauerhaft und korrespondieren zeitlich nicht exakt mit dem Sensorausfall. Die Fehlalarmraten für $FAR_{T^2} = 0,3\%$ und $FAR_{SPE} = 1,4\%$ zeigen, dass es selbst im fehlerfreien Zustand zu relativ häufigen Fehlalarmen kommt. Eine Optimierung

insbesondere des *SPE*-Schwellwertes könnte die Anzahl der Fehlalarme senken. Der Ausschlag der Amplitude des *SPE*-Indexes ist zwar deutlich, aber im Vergleich zu den beiden vorangegangenen Fehlern nicht so mächtig.

7.1.4 Erhöhte Schwingungen an der Antriebstrommel 2

Die Zeitverläufe der in den Abbildungen 7.12 und 7.13 dargestellten Messgrößen zeigen, dass die erhöhten Schwingungen an dem Messgeber der Antriebstrommel 2 ab dem 12.03.2009 auftraten. Eine direkte Beeinflussung anderer Messgrößen konnte nicht festgestellt werden.

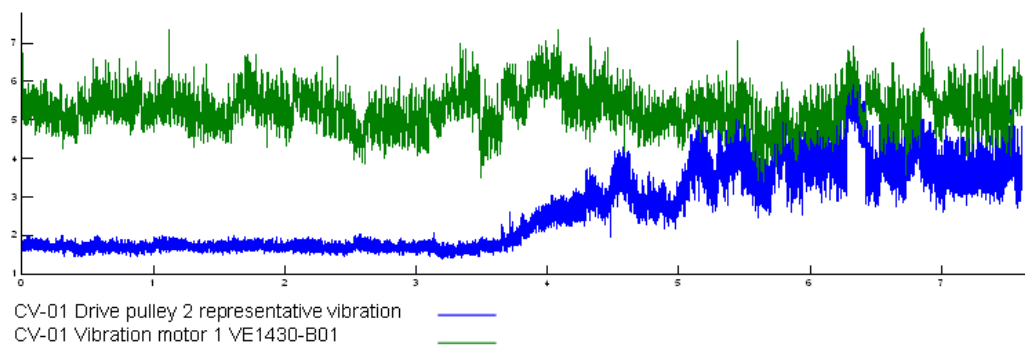


Abb. 7.12: Fehler Erhöhte Schwingungen an Antriebstrommel 2: Vibrationen an den Motoren der Antriebstrommeln 1 und 2

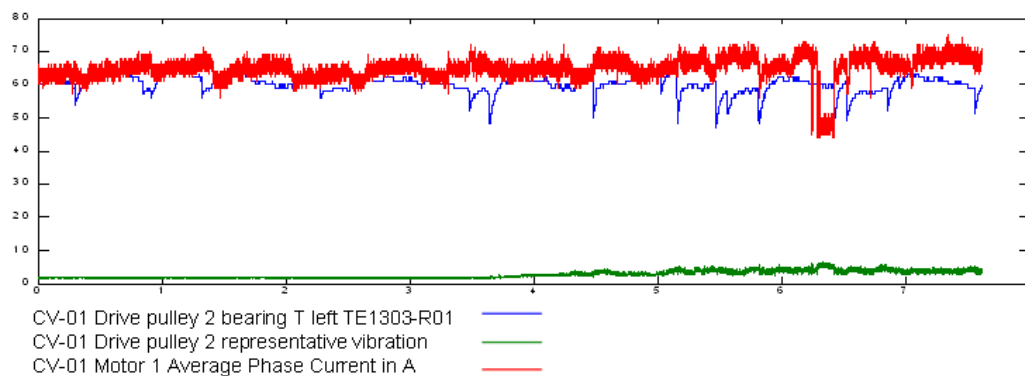


Abb. 7.13: Fehler Erhöhte Schwingungen an Antriebstrommel 2: Strom Motor 1, Temperatur und Vibration Antriebstrommel 2

Das Normalverhalten wurde über die Ersten 30000 Sample des verfügbaren Datensatzes angelernt. Die Software bestimmte mithilfe des Kaiser-Kriteriums 4 Eigenvektoren zu Hauptkomponenten. Die Fehlalarmraten für die Daten des Normalzustandes liegen für den T^2 -Index bei $FAR_{T^2} = 0,37\%$ sowie für den *SPE*-Index bei $FAR_{SPE} = 0,78\%$, sind zwar geringer als bei den vorherigen Untersuchungen, aber für den praktischen Einsatz noch zu hoch.

Nachfolgende Abbildungen zeigen die berechneten Indizes für die Daten dieses Fehlerbildes.

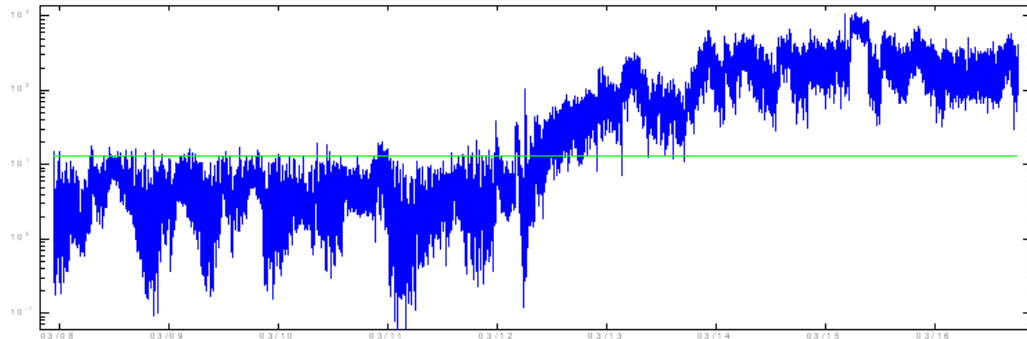


Abb. 7.14: Fehler Erhöhte Schwingungen an Antriebstrommel 2: T^2 -Index nach der PCA

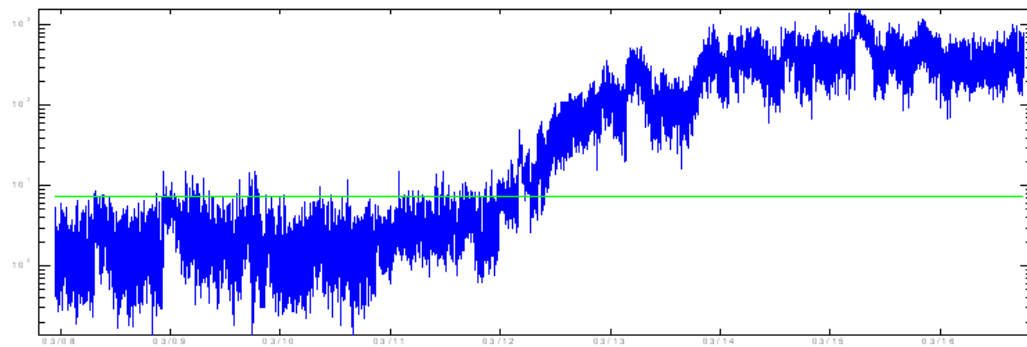


Abb. 7.15: Fehler Erhöhte Schwingungen an Antriebstrommel 2: SPE -Index nach der PCA

Sowohl der T^2 - als auch der SPE -Index, dargestellt in den Abbildungen 7.14 und 7.15, überschreitet deutlich den, über die Daten des Normalzustandes bestimmten, Schwellwert. Der Fehler kann somit sicher detektiert werden.

7.2 Datenvorverarbeitung

7.2.1 Verminderung von Dynamiken im Erwartungswert durch lineare Kombination von Messgrößen

Die Fehlerdetektion mittels der PCA wird durch Dynamiken im Erwartungswert wesentlich beeinflusst. In diesem Abschnitt soll gezeigt werden, dass sich durch die gezielte Vorverarbeitung von Daten der Gurtförderanlage CV01 mittels eines Verfahrens der linearen Kombination von Messgrößen, Dynamiken im Erwartungswert reduzieren lassen und Abweichungen zwischen diesen Größen mittels der Fehlerdetektion durch die PCA besser erkannt werden.

Dazu wurden Daten der Gurtförderanlage CV-01 mittels der PCA verarbeitet. Schwerpunkt der Untersuchung waren die Schwingungen an der linken und rechten Seite der Antriebstrommel 1. Der Signalverlauf dieser beiden Messgrößen ist im Diagramm der Abbildung 7.16 zu sehen. Darüber hinaus wurde von beiden Größen das Differenzsignal gebildet, welches ebenfalls im Diagramm der Abbildung 7.16 dargestellt ist. Die ab Sample 3500 erkennbaren Abweichungen zwischen beiden Schwingungssignalen sollen mithilfe der linearen Kombination beider Signale und der nachfolgenden Fehlerdetektion mittels PCA deutlich in den Indizes angezeigt werden.

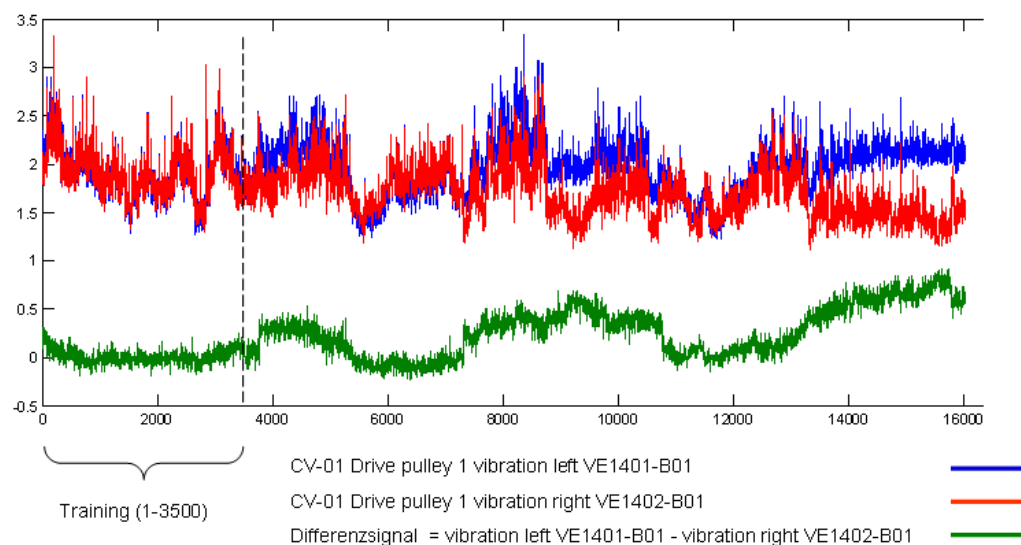


Abb. 7.16: Schwingungen Antriebstrommel links und rechts sowie Differenz zwischen beiden Signalen

Da die rechte und linke Seite der Antriebstrommel über eine kompakte, relativ starre Welle verbunden sind, ist ein eindeutig definierbarer physikalischer Zusammenhang zwischen dem rechten und linken Schwingungssignal beschreibbar. Das zeigt sich ebenfalls in der Größe des Korrelationskoeffizienten beider Messgrößen (Signal 2 und Signal 3), wie man der folgenden Abbildung 7.17 entnehmen kann. In dieser Tabelle sind die über alle 19 Messgrößen gebildeten Korrelationskoeffizienten enthalten, welche zur Fehlerdetektion mittels der PCA herangezogen wurden.

7 Ergebnisse

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Drive pulley 1 representative vibration	1.00	0.99	0.88	-0.14	-0.16	0.01	0.11	0.45	0.45	0.45	0.45	0.47	-0.29	0.58	-0.46	-0.32	0.58	0.61	0.71
Drive pulley 1 vibration left VE1401-B01	0.99	1.00	0.84	-0.16	-0.18	0.02	0.12	0.49	0.49	0.49	0.49	0.51	-0.31	0.62	-0.50	-0.35	0.62	0.65	0.73
Drive pulley 1 vibration right VE1402-B01	0.88	0.84	1.00	0.07	0.07	0.00	0.03	0.17	0.17	0.17	0.17	0.18	0.03	0.30	-0.16	-0.05	0.30	0.33	0.58
Drive pulley 2 bearing T left TE1303-R01	-0.14	-0.16	0.07	1.00	0.95	0.02	-0.14	-0.53	-0.53	-0.53	-0.52	-0.54	0.84	-0.57	0.79	0.91	-0.57	-0.38	-0.17
Drive pulley 2 bearing T right TE1304-R02	-0.16	-0.18	0.07	0.95	1.00	0.02	-0.19	-0.56	-0.55	-0.56	-0.55	-0.56	0.91	-0.59	0.78	0.90	-0.59	-0.39	-0.21
Drive pulley 2 representative vibration	0.01	0.02	0.00	0.02	0.02	1.00	0.81	0.01	0.01	0.01	0.01	0.01	0.00	0.05	-0.03	-0.02	0.05	0.10	0.09
Drive pulley 2 vibration left VE1403-B01	0.11	0.12	0.03	-0.14	-0.19	0.81	1.00	0.14	0.14	0.14	0.14	0.14	-0.26	0.19	-0.16	-0.18	0.19	0.19	0.21
Motor 1 Average Phase Current in A	0.45	0.49	0.17	-0.53	-0.56	0.01	0.14	1.00	0.95	1.00	0.94	0.86	-0.67	0.75	-0.76	-0.68	0.75	0.65	0.43
Motor 1 Phase A Current in A	0.45	0.49	0.17	-0.53	-0.55	0.01	0.14	0.95	1.00	0.95	0.95	0.85	-0.66	0.75	-0.76	-0.67	0.75	0.66	0.43
Motor 1 Phase B Current in A	0.45	0.49	0.17	-0.53	-0.56	0.01	0.14	1.00	0.95	1.00	0.94	0.86	-0.67	0.75	-0.76	-0.68	0.75	0.65	0.43
Motor 1 Phase C Current in A	0.45	0.49	0.17	-0.52	-0.55	0.01	0.14	0.94	0.95	0.94	1.00	0.86	-0.67	0.75	-0.76	-0.67	0.75	0.65	0.43
Motor 1 Real Power in kW	0.47	0.51	0.18	-0.54	-0.56	0.01	0.14	0.86	0.85	0.86	0.86	1.00	-0.67	0.76	-0.78	-0.69	0.76	0.67	0.44
Motor 1 gearbox temperature TE1321-R01	-0.29	-0.31	0.03	0.84	0.91	0.00	-0.26	-0.67	-0.66	-0.67	-0.67	-0.67	1.00	-0.69	0.80	0.87	-0.69	-0.54	-0.31
Representative vibration gearbox motor 1	0.58	0.62	0.30	-0.57	-0.59	0.05	0.19	0.75	0.75	0.75	0.75	0.76	-0.69	1.00	-0.87	-0.77	1.00	0.88	0.63
Representative bearing T Motor 1	-0.46	-0.50	-0.16	0.79	0.78	-0.03	-0.16	-0.76	-0.76	-0.76	-0.76	-0.78	0.80	-0.87	1.00	0.95	-0.87	-0.73	-0.49
Representative winding T Motor 1	-0.32	-0.35	-0.05	0.91	0.90	-0.02	-0.18	-0.68	-0.67	-0.68	-0.67	-0.69	0.87	-0.77	0.95	1.00	-0.77	-0.59	-0.37
Vibration gearbox 1 Sensor 1 VE1421-B01	0.58	0.62	0.30	-0.57	-0.59	0.05	0.19	0.75	0.75	0.75	0.75	0.76	-0.69	1.00	-0.87	-0.77	1.00	0.88	0.63
Vibration gearbox 1 Sensor 2 VE1422-B01	0.61	0.65	0.33	-0.38	-0.39	0.10	0.19	0.65	0.66	0.65	0.65	0.67	-0.54	0.88	-0.73	-0.59	0.88	1.00	0.68
Vibration motor 1 VE1411-B01	0.71	0.73	0.58	-0.17	-0.21	0.09	0.21	0.43	0.43	0.43	0.43	0.44	-0.31	0.63	-0.49	-0.37	0.63	0.68	1.00

Abb. 7.17: Korrelationsmatrix gebildet aus den Trainingsdaten

Im nächsten Abschnitt wird das Differenzsignal zunächst nur zwischen den beiden Schwingungssignalen der rechten und linken Seite der Antriebstrommel 1 gebildet und als zusätzliche Information für die Fehlerdetektion benutzt.

7.2.1.1 PCA mit einem Differenzsignal

Wie bereits beschrieben, wurde vor der Anwendung des PCA-Verfahrens das Differenzsignal aus den linken und rechten Schwingungsmessgrößen der Antriebstrommel (Datenspur 2 und 3) erzeugt. Dieses ersetzt nun das Schwingungssignal rechts (Datenspur 3). Da das Schwingungssignal links im Datensatz unverändert erhalten bleibt, besteht auch weiterhin die Möglichkeit, Veränderungen im Erwartungswert dieses Messsignals zu detektieren. Anschließend erfolgt die Normierung der Daten, die PCA und die Bestimmung der T^2 - und SPE -Indizes. Die Abbildungen 7.18 und 7.19 zeigen die mittels der PCA ermittelten Eigenwerte, einerseits ohne Differenzbildung und andererseits mit Differenzbildung. Die Eigenvektoren mit den vier größten Eigenwerten wurden nach dem Kaiser-Kriterium als Hauptkomponenten bestimmt.

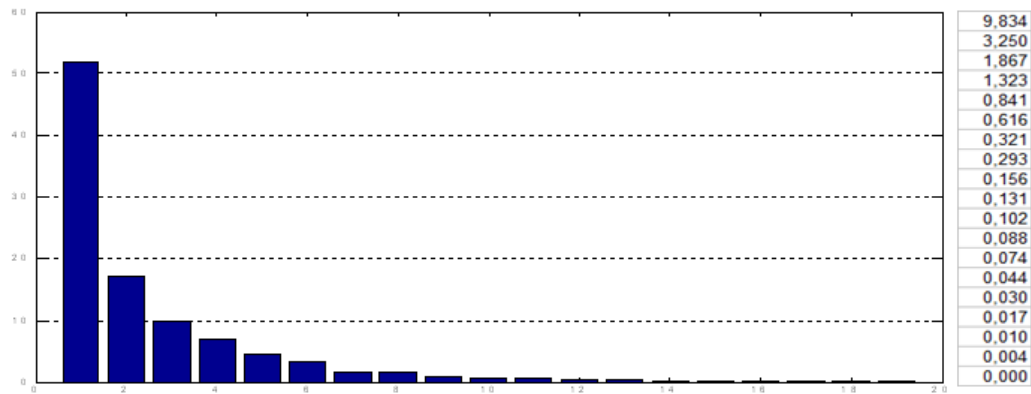


Abb. 7.18: Eigenwerte ohne Differenzbildung zwischen den Schwingungssignalen links und rechts, prozentual und absolut

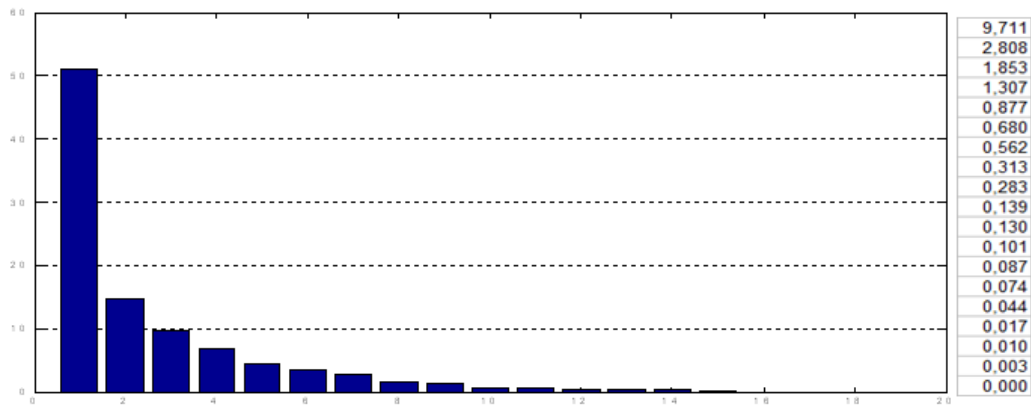


Abb. 7.19: Eigenwerte nach der Differenzbildung zwischen den Schwingungssignalen links und rechts, prozentual und absolut

Bei genauer Betrachtung der Eigenwerte ist erkennbar, dass durch die Differenzbildung die Beträge der größten Eigenwerte etwas sinken, während die Beträge der kleineren Eigenwerte etwas steigen gegenüber den Eigenwerten ohne Differenzbildung. Das bestätigt die Untersuchungen des Abschnittes 4.2.1. Nachfolgende Abbildungen zeigen die mittels T^2 - und SPE -Statistik gebildeten Indizes ohne und mit Differenzbildung.

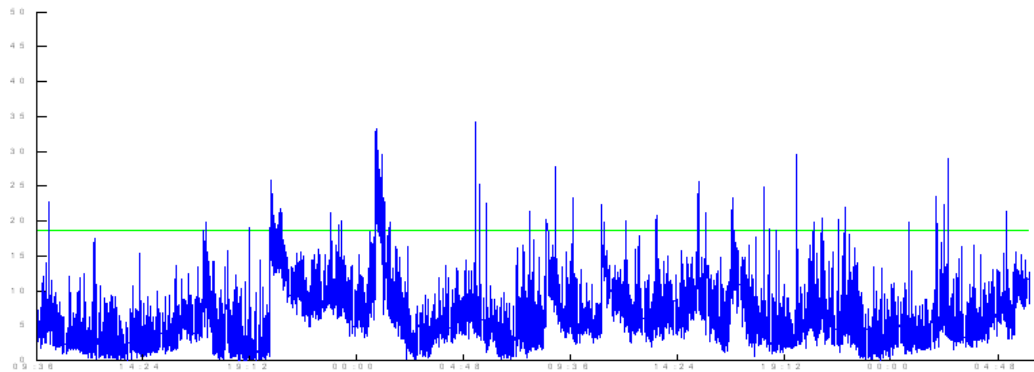


Abb. 7.20: T^2 -Index ohne Differenzbildung

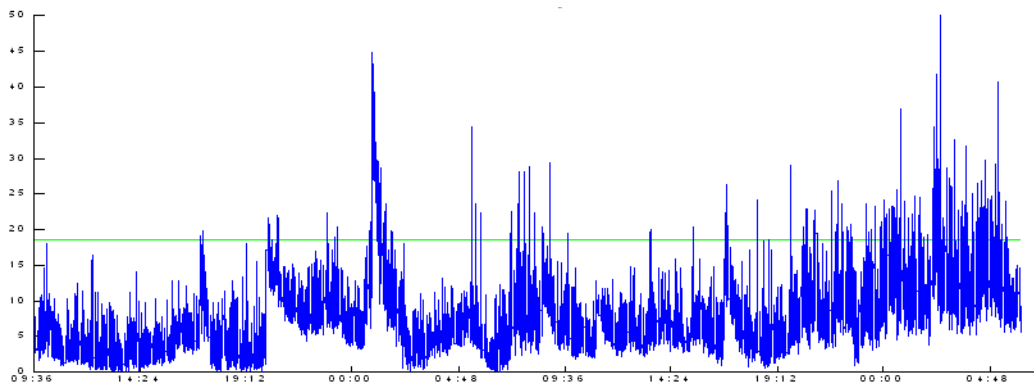


Abb. 7.21: T^2 -Index nach der Differenzbildung zwischen den Schwingungssignalen links und rechts

Die Auswertung der T^2 -Indizes ohne die Differenzbildung in Abbildung 7.20 zeigt, dass die Abweichungen zwischen linker und rechter Trommel mittels T^2 -Statistik nicht bestimmbar sind. Erst nach der Differenzbildung ist eine leichte Verbesserung sichtbar. Für eine zuverlässige Fehlerdetektion ist das aber nicht ausreichend.

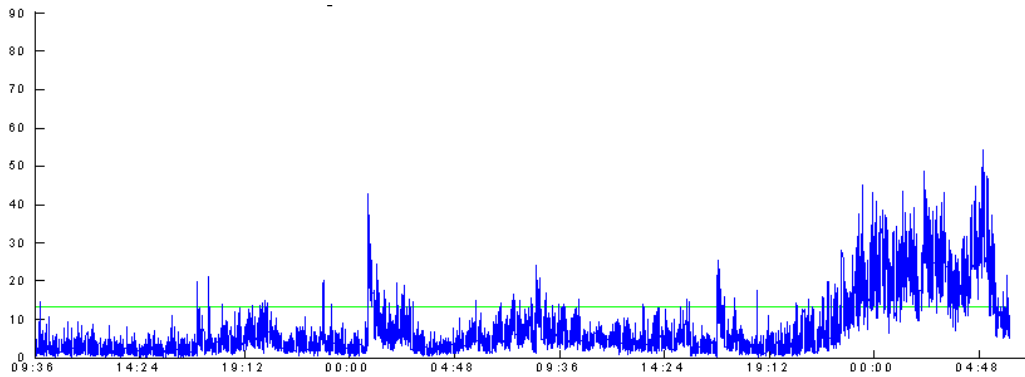


Abb. 7.22: *SPE*-Index ohne Differenzbildung

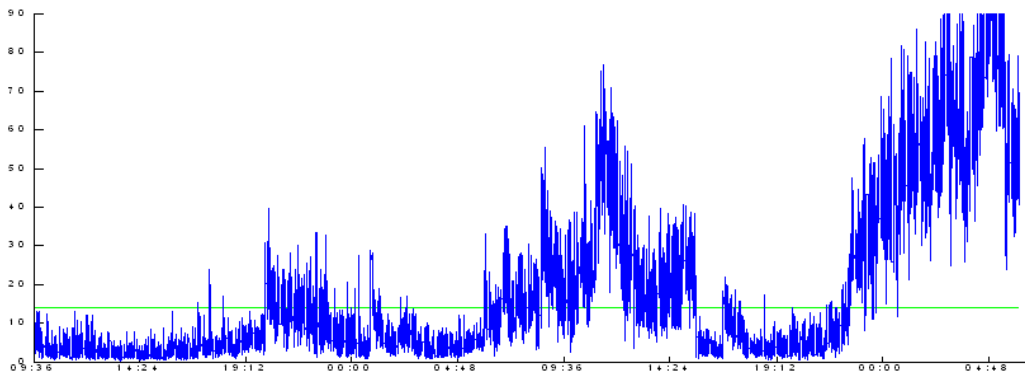


Abb. 7.23: *SPE*-Index nach der Differenzbildung zwischen den Schwingungssignalen links und rechts

Im Gegensatz zu den T^2 -Indizes ist bei den *SPE*-Indizes durch die Differenzbildung eine signifikante Verbesserung der Detektierbarkeit der Abweichungen zwischen den linken und rechten Schwingungssignalen der Antriebstrommel 1 erkennbar. Die Verkleinerung der deterministischen Anteile im Erwartungswert führt zu einem größeren Einfluss der zufälligen Anteile (Fehleranteile), welche insbesondere im Residuum-Unterraum vorhanden sind. Somit bestätigen die Abbildungen mit den *SPE*-Indizes (7.22 und 7.23) die im Abschnitt 4.2.1 getätigten Aussagen, dass durch die lineare Kombination von Messwerten eine verbesserte Detektion von zufälligen Einflussgrößen (Fehlern) möglich ist und sich diese insbesondere im *SPE*-Index äußern.

7.2.1.2 PCA mit Differenzsignalen aus allen stark korrelierenden Signalen

Nach der linearen Kombination von zwei Messgrößen sollte in diesem Schritt gezeigt werden, ob durch die Differenzbildung über alle Messgrößen zwischen denen ein starker Zusammenhang besteht, weitere Verbesserungen bezüglich der Detektions-Performanz erzielbar sind. Den Ausgangspunkt bildet wieder die Korrelationsmatrix aus den Trainingsdaten, siehe Abbildung 7.24.

7 Ergebnisse

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1.00	0.99	0.88	-0.14	-0.16	0.01	0.11	0.45	0.45	0.45	0.45	0.47	-0.29	0.58	-0.46	-0.32	0.58	0.61	0.71
1-2	0.99	1.00	0.84	-0.16	-0.18	0.02	0.12	0.49	0.49	0.49	0.49	0.51	-0.31	0.62	-0.50	-0.35	0.62	0.65	0.73
1-3	0.88	0.84	1.00	0.07	0.07	0.00	0.03	0.17	0.17	0.17	0.17	0.18	0.03	0.30	-0.16	-0.05	0.30	0.33	0.58
4	-0.14	-0.16	0.07	1.00	0.95	0.02	-0.14	-0.53	-0.53	-0.53	-0.52	-0.54	0.84	-0.57	0.79	0.91	-0.57	-0.38	-0.17
4-5	-0.16	-0.18	0.07	0.95	1.00	0.02	-0.19	-0.56	-0.55	-0.56	-0.55	-0.56	0.91	-0.59	0.78	0.90	-0.59	-0.39	-0.21
4-13	0.01	0.02	0.00	0.02	0.02	1.00	0.81	0.01	0.01	0.01	0.01	0.01	0.00	0.05	-0.03	-0.02	0.05	0.10	0.09
4-15	0.11	0.12	0.03	-0.14	-0.19	0.81	1.00	0.14	0.14	0.14	0.14	0.14	-0.26	0.19	-0.16	-0.18	0.19	0.19	0.21
4-16	0.45	0.49	0.17	-0.53	-0.56	0.01	0.14	1.00	0.95	1.00	0.94	0.86	-0.67	0.75	-0.76	-0.68	0.75	0.65	0.43
6	0.45	0.49	0.17	-0.53	-0.55	0.01	0.14	0.95	1.00	0.95	0.95	0.85	-0.66	0.75	-0.76	-0.67	0.75	0.66	0.43
6-7	0.45	0.49	0.17	-0.53	-0.56	0.01	0.14	1.00	0.95	1.00	0.94	0.86	-0.67	0.75	-0.76	-0.68	0.75	0.65	0.43
8	0.45	0.49	0.17	-0.52	-0.55	0.01	0.14	0.94	0.95	0.94	1.00	0.86	-0.67	0.75	-0.76	-0.67	0.75	0.65	0.43
8-9	0.47	0.51	0.18	-0.54	-0.56	0.01	0.14	0.86	0.85	0.86	0.86	1.00	-0.67	0.76	-0.78	-0.69	0.76	0.67	0.44
8-10	-0.29	-0.31	0.03	0.84	0.91	0.00	-0.26	-0.67	-0.66	-0.67	-0.67	-0.67	1.00	-0.69	0.80	0.87	-0.69	-0.54	-0.31
8-11	0.58	0.62	0.30	-0.57	-0.59	0.05	0.19	0.75	0.75	0.75	0.75	0.76	-0.69	1.00	-0.87	-0.77	1.00	0.88	0.63
8-12	-0.46	-0.50	-0.16	0.79	0.78	-0.03	-0.16	-0.76	-0.76	-0.76	-0.76	-0.78	0.80	-0.87	1.00	0.95	-0.87	-0.73	-0.49
14	-0.32	-0.35	-0.05	0.91	0.90	-0.02	-0.18	-0.68	-0.67	-0.68	-0.67	-0.69	0.87	-0.77	0.95	1.00	-0.77	-0.59	-0.37
14-17	0.58	0.62	0.30	-0.57	-0.59	0.05	0.19	0.75	0.75	0.75	0.75	0.76	-0.69	1.00	-0.87	-0.77	1.00	0.88	0.63
14-18	0.61	0.65	0.33	-0.38	-0.39	0.10	0.19	0.65	0.66	0.65	0.65	0.67	-0.54	0.88	-0.73	-0.59	0.88	1.00	0.68
19	0.71	0.73	0.58	-0.17	-0.21	0.09	0.21	0.43	0.43	0.43	0.43	0.44	-0.31	0.63	-0.49	-0.37	0.63	0.68	1.00

Abb. 7.24: Korrelationsmatrix gebildet aus den Trainingsdaten

Im nächsten Schritt wurden zwischen allen Messgrößen, die einen Korrelationskoeffizienten von größer 0,8 besaßen, das Differenzsignal erzeugt. Jedoch blieb, wie im vorigen Abschnitt, immer ein Originalsignal erhalten. Anschließend erfolgte die Normierung der Daten, die PCA sowie die Berechnung der T^2 - und SPE -Indizes. Die nachfolgenden Abbildungen zeigen die Ergebnisse der Berechnungen.

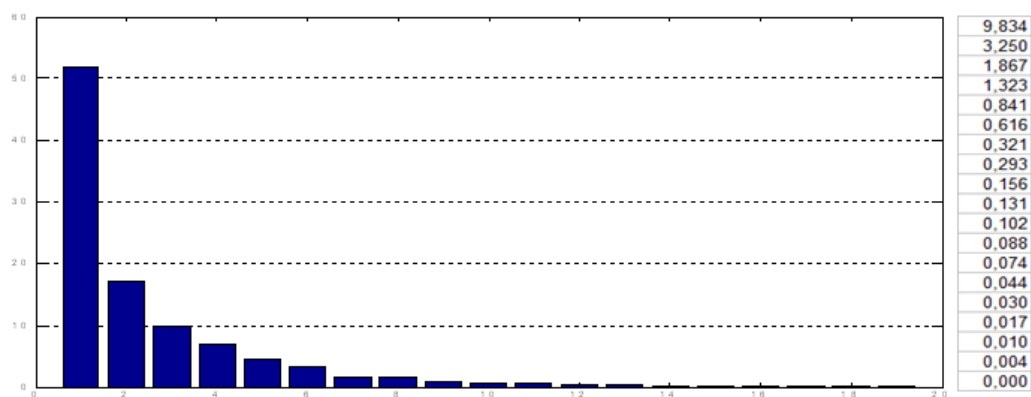


Abb. 7.25: Eigenwerte ohne Differenzbildung

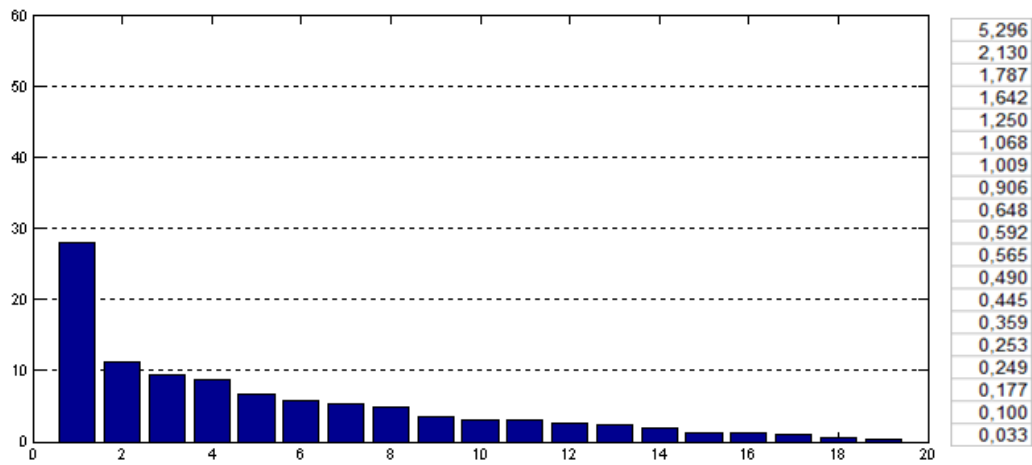


Abb. 7.26: Eigenwerte nach der Differenzbildung zwischen mehreren Signalen

Vergleicht man die berechneten Eigenwerte aus den Abbildungen 7.25 und 7.26, so sind hier noch deutlicher, als in dem Beispiel mit einem Differenzsignal, die unterschiedlichen prozentualen Verteilungen sichtbar. Die Eigenwerte des Residuum-Unterraumes sind nach der Differenzbildung deutlich stärker ausgeprägt. Über das Kaiser-Kriterium wurden nun sieben Eigenvektoren zu Hauptkomponenten bestimmt. Die folgenden Abbildungen enthalten die berechneten T^2 - und SPE -Indizes ohne und mit Differenzbildung.

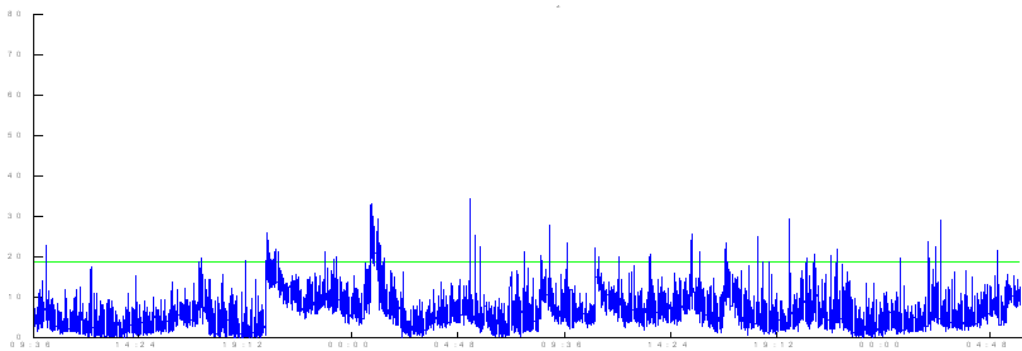


Abb. 7.27: T^2 -Index ohne Differenzbildung

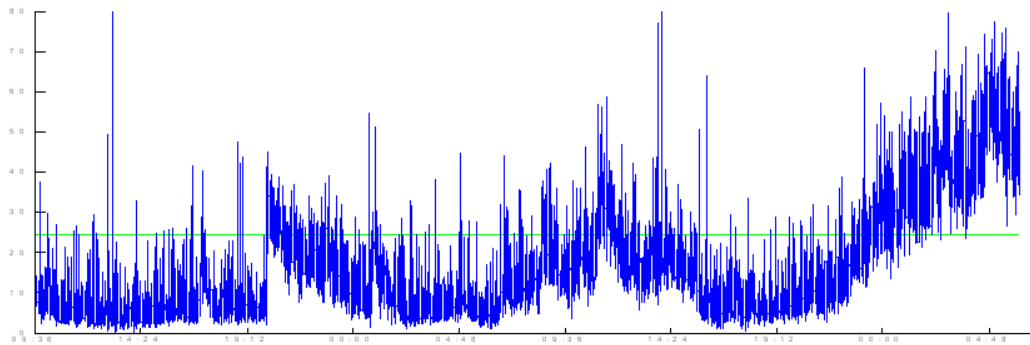


Abb. 7.28: T^2 -Index nach der Differenzbildung zwischen mehreren Signalen

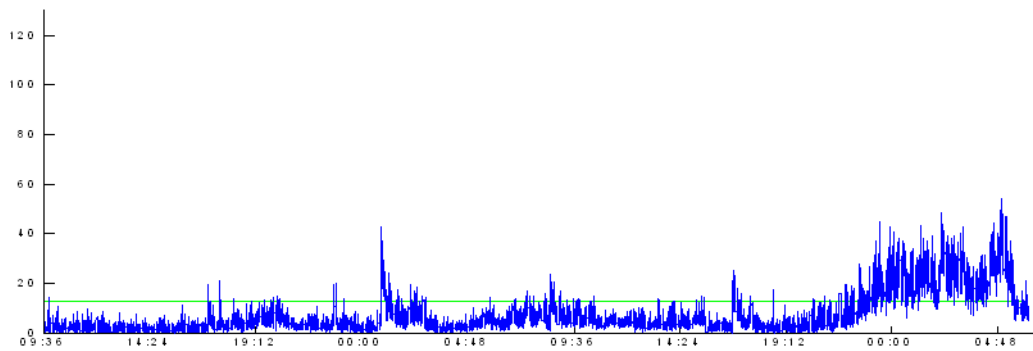


Abb. 7.29: SPE -Index ohne Differenzbildung

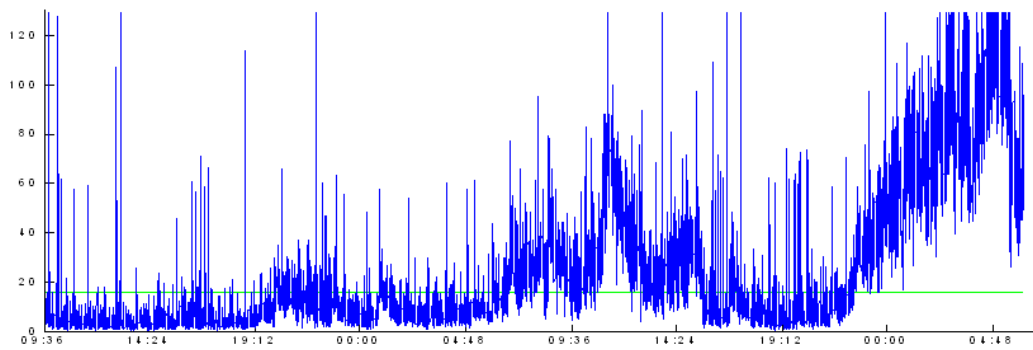


Abb. 7.30: SPE -Index nach der Differenzbildung

Vergleicht man nun die berechneten T^2 - und SPE -Indizes ohne und mit Differenzbildung, so ist augenscheinlich, dass die Indizes mit Differenzbildung deutlich sensibler auf Abweichungen reagieren. Die große Empfindlichkeit geht aber einher mit großen Spitzen in den Indizes, was zu einer hohen Fehlalarmrate im fehlerfreien Bereich führt. So angewendet eignet sich das Verfahren nicht zur Fehlerdetektion. Die großen Ausschläge haben ihre Ursachen unter anderem in Abweichungen zwischen stark korrelierenden Signalen. Diese entstehen nicht zufällig, sondern finden ihre Begründungen in den physikalischen Zusammenhängen zwischen den Messgrößen, welche durch die Differenzbildung unzureichend erklärt wurden. Hier könnten andere Verfahren zur linearen Kombination oder auch Subspace-Verfahren bessere Ergebnisse liefern. Diese Untersuchungen sind aber nicht Gegenstand dieser Arbeit, sondern bedürfen weiterer Forschungen.

7.2.2 Anpassung der Varianz

Wie im Abschnitt 4.2.2 beschrieben, kann man die Normalverteilung der Messdaten nicht immer als gegeben annehmen. Bereits kleine Instationaritäten oder auch transiente Zustände führen zu Fehlalarmen. Werden die transienten Prozesszustände anhand eindeutiger Merkmale erkannt, können sie zum Beispiel off-line herausgeschnitten werden oder die on-line Fehlerdetektion stoppt für diese Bereiche oder die Daten dieser Bereiche werden mit einem anderen statistischen Verfahren untersucht. Damit lassen sich aber die im Abschnitt 4.2.2 beschriebenen instationären Bereiche nicht vollständig ausblenden.

Die vorgeschlagene Anpassung der empirischen Kovarianzmatrix (des „statistischen Modells“) soll das Fehlerdetektionssystem unempfindlicher bezüglich kleiner Instationaritäten machen. Dazu ersetzt man die aus den nicht normalverteilten Daten gebildete Kovarianzmatrix, durch eine „künstliche“ Kovarianzmatrix, gebildet aus multinormalverteilten Daten mit den gleichen statistischen Parametern (Mittelwert und Standardabweichung).

Nachfolgend wird der bereits im Abschnitt 7.1 mit dem PCA-Verfahren ausgewertete Fehler „Wellenbruch an Antriebsstrommel 1“ erneut mit der PCA aber diesmal mit Anpassung der Varianz untersucht. Auf Basis der im off-line-Training ermittelten Mittelwerte und Standardabweichungen wurden neue normalverteilte Datenreihen erzeugt, deren neue Standardabweichungen jedoch das 1,5-fache der ursprünglichen betragen. Damit sollte eine größere Robustheit gegenüber Fehlalarmen erreicht werden. Nach der Normierung der neu gebildeten Datenmatrix X wurde anschließend die Kovarianzmatrix gebildet und danach die Singulärwertzerlegung durchgeführt. Die nachfolgenden Balkendiagramme zeigen die Eigenwerte, einmal aus der Kovarianzmatrix der originalen Trainingsdaten und im Vergleich dazu die Eigenwerte aus der Kovarianzmatrix der künstlich erzeugten normalverteilten Daten.

Betrachtet man beide Diagramme fällt beim zweiten Diagramm die relativ homogene Verteilung der Eigenwertbeträge auf alle Eigenvektoren auf. Die Ursachen dafür liegen in der Normalverteilung der Daten und den folglich geringen Kovarianzen. Die

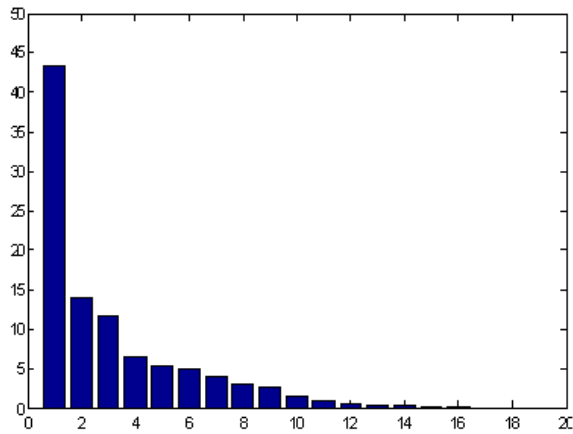


Abb. 7.31: Fehler: Wellenbruch an Antriebstrommel 1; Eigenwerte

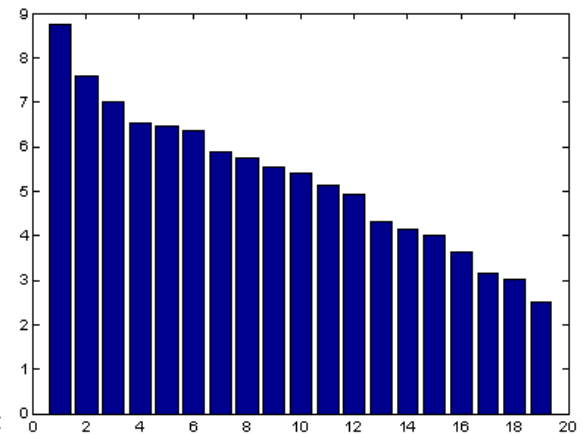


Abb. 7.32: Fehler: Wellenbruch an Antriebstrommel 1; Eigenwerte nach Neubildung der Kovarianzmatrix

beiden Abbildungen 7.33 und 7.34 zeigen den T^2 - und SPE -Index nach der on-line-Überwachung mit den neu gebildeten Eigenwerten. Wegen der großen Index-Werte im Fehlerfall wurde eine logarithmische Einteilung der Y-Achse gewählt.

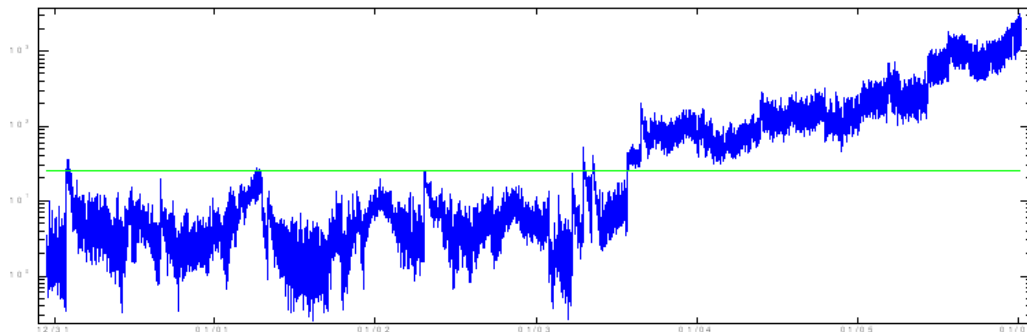


Abb. 7.33: Fehler: Wellenbruch an Antriebstrommel 1; T^2 -Index nach Neubildung der Kovarianzmatrix

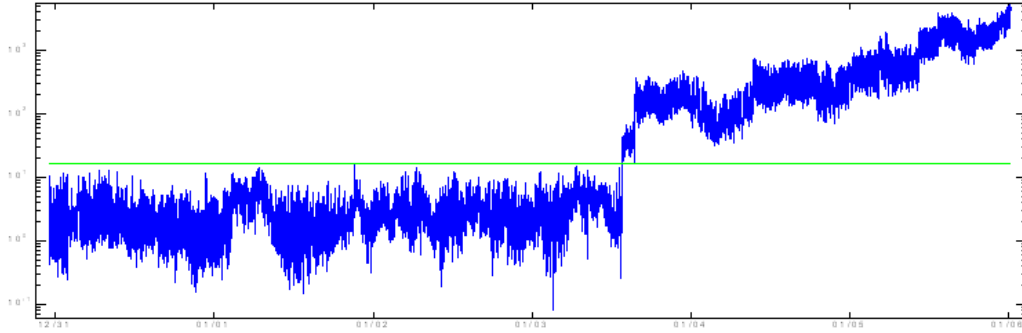


Abb. 7.34: Fehler: Wellenbruch an Antriebstrommel 1; *SPE*-Index nach Neubildung der Kovarianzmatrix

Im Vergleich zu den Ergebnissen der Standard-PCA 7.1 beträgt die Fehlalarmraten für den T^2 -Index $FAR_{T^2} = 0,12\%$ und für den *SPE*-Index sogar nur $FAR_{SPE} = 0,004\%$. Sie konnte gegenüber der Standard-PCA deutlich gesenkt werden, wobei die Indizes mit Werten zwischen 200 und 30000 im Fehlerfall noch ausreichend deutlich über den Schwellwerten liegen. Somit bestand auch keine Notwendigkeit, die Algorithmen zur Bildung der Schwellwerte anzupassen. Der Fehler wurde wie bei der Standard-PCA rechtzeitig und sicher detektiert. Für die nachfolgend beschriebenen Ergebnisse der Fehlerdetektion mithilfe der Methode des GLR hatte diese Anpassung eine noch größere Bedeutung.

7.3 Fehlerdetektion mithilfe der Methode des GLR

Die im Abschnitt 6.3 beschriebenen vier Fehlerzustände wurden nachfolgend mit den im Abschnitt 3.3.5 erläuterten Verfahren zur Fehlerdetektion mithilfe des Generalized Likelihood Ratio Tests (GLRT) untersucht.

7.3.1 Ausfall der Antriebstrommel 1 durch Wellenbruch

Zunächst erfolgte die Auswertung der Daten zum Wellenbruch an der Antriebstrommel 1. Für das off-line-Training wurden die Sample 1 bis 25000 normiert, die Kovarianzmatrix gebildet sowie der Schwellwert J_{Th} nach Formel 3.72 mit einem Vertrauensbereich von 99,9% bestimmt. Es folgte die Berechnung der Indizes und die Auswertung entsprechend der Logik der Formel 3.73.

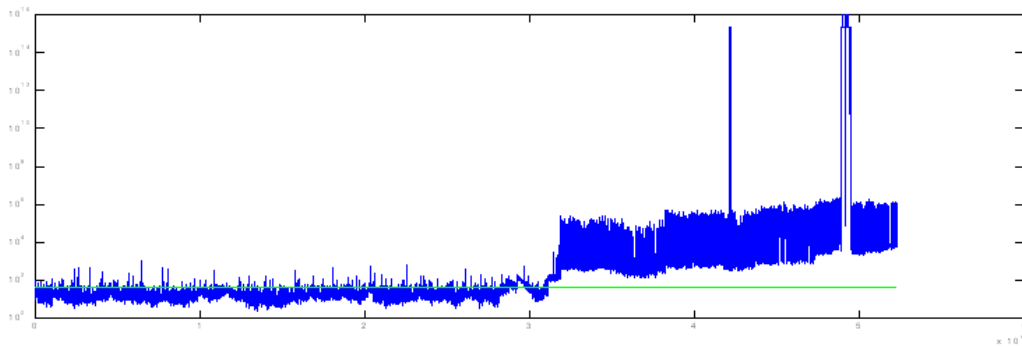


Abb. 7.35: Fehler: Wellenbruch an Antriebstrommel 1; T^2 -Index nach GLR

Die Abbildung 7.35 zeigt den Verlauf des T^2 -Index und den zugehörigen Schwellwert. Auffällig sind die starken Überschreitungen des Schwellwertes im Trainings- und die extrem großen Index-Werte im Fehlerbereich. Damit man die relativ kleinen Indizes der Trainingsdaten zusammen mit den großen Indizes der Fehlerdaten vernünftig in einem Diagramm betrachten kann, wurde eine logarithmische Darstellung gewählt. Die starken Schwankungen in den Indizes lassen auf eine sehr hohe Empfindlichkeit des T^2 -Index nach der GLR-Methode gegenüber den Indizes nach dem PCA-Verfahren schließen. Die Ursachen dafür kann man über die bei der GLR-Methode noch vorhandenen Kovarianzen erklären. Bei der PCA wird zur T^2 -Index-Bildung durch die Eigenwerte der Hauptkomponenten dividiert, welche nur die größten Varianzanteile enthalten. Im Unterschied dazu dividiert man bei der GLR-Methode durch die Varianzen und Kovarianzen.

Für den Trainingsbereich führen die im Abschnitt 4.2.2 beschriebenen kleinen Instationaritäten infolge der großen Empfindlichkeit zu einer sehr hohen Fehlalarmrate von $FAR_{T^2} = 2,65\%$. Selbst eine Vergrößerung des Vertrauensbereiches auf einen Wert von 99,9937% und damit auf das 4-fache der Standardabweichung σ brachte keine deutliche Verbesserung. Der Einsatz des Fehlerdetektionsverfahrens mit der Methode des GLR in der Industrie ist unter diesen Umständen nicht praktikabel. Deshalb wurden folgende zwei Veränderungen vorgenommen:

1. Für die Minimierung der Fehlalarmrate im fehlerfreien Bereich wurde ein neuer sicherer Schwellwert bestimmt. Dazu ermittelt man während des off-line-Trainings zunächst den maximalen Wert des jeweiligen Indexes über den Trainingsbereich. Der neue Schwellwert ist das 1,1-fache des maximalen Indexwertes.
2. Ist die Fehlalarmrate im Trainingsbereich bei einem Vertrauensbereich von 99,9937% größer als 1%, so erfolgt zusätzlich die im Abschnitt 4.2.2 beschriebene Anpassung der Kovarianzmatrix. Liegt ein Index-Wert außerhalb des Vertrauensbereiches von 99,9937%, so hat er den Mittelwert um das 4-fache der Standardabweichung σ überschritten.

Die nachfolgende Abbildung 7.36 zeigt zunächst den T^2 -Index mit dem neuen Schwellwert nach Punkt 1, aber ohne die Ausführung von Punkt 2.

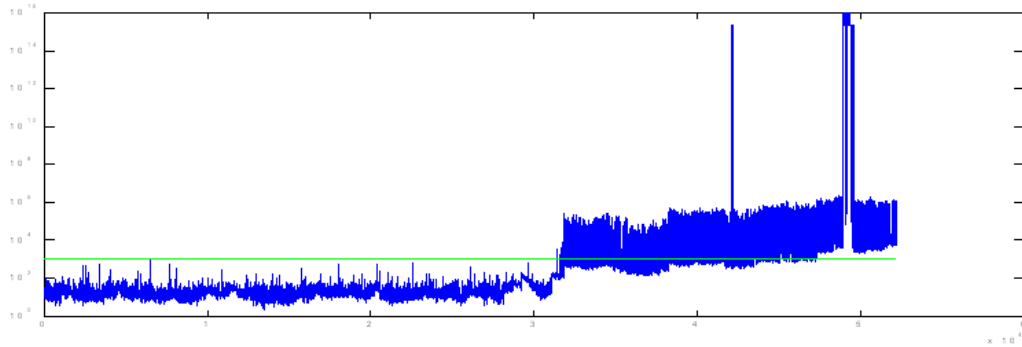


Abb. 7.36: Fehler Wellenbruch an Antriebstrommel 1; T^2 -Index nach GLR; Schwellwert aus maximalem T^2 -Index der Trainingsdaten

Zwar überschritten die Indizes im Trainingsbereich nun nicht mehr den Schwellwert, aber dafür war dieser nun so groß, dass es im Fehlerfall zu vielen Unterschreitungen des Schwellwertes kam. Da aber die Indizes im Fehlerfall immer noch Werte von 10^{16} erreichten, erfolgte anschließend eine Anpassung der Kovarianzmatrix nach Punkt 2. Die Resultate zeigt die nachfolgende Abbildung 7.37.

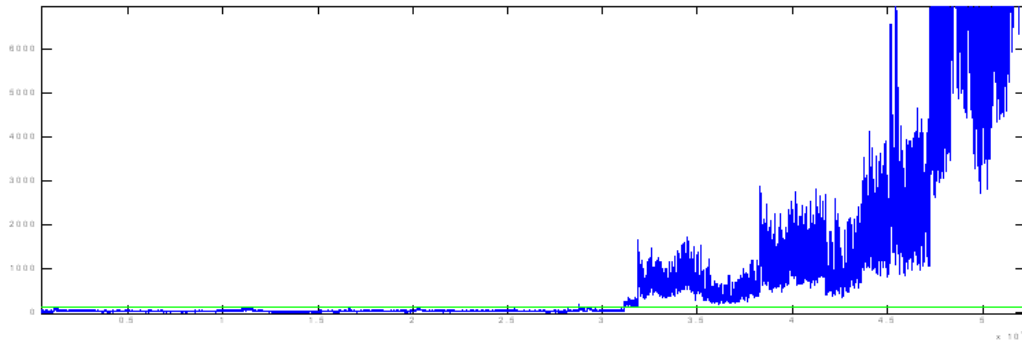


Abb. 7.37: Fehler: Wellenbruch an Antriebstrommel 1; T^2 -Index nach GLR; Schwellwert aus maximalem T^2 -Index der Trainingsdaten und angepasster Kovarianzmatrix

Nach der Anpassung der Varianz nach Punkt 2 sank die Empfindlichkeit des T^2 -Indexes augenfällig. Für die Anzeige im Diagramm konnte deshalb auf eine logarithmische Darstellung verzichtet werden. Der Schwellwert liegt über den Indizes der Trainingsdaten, aber deutlich unter den Fehlerdaten. Der Fehlerzustand wurde nun sicher und rechtzeitig angezeigt.

7.3.2 Mantelriss

Das off-line Training für die Untersuchung der Daten zum Mantelriss erfolgte wie bei der PCA über die Sample 1 bis 15000. Die nachfolgende Abbildung zeigt den T^2 -Index und den Schwellwert J_{Th} laut Formel 3.72 mit einem Vertrauensbereich von 99,9%.

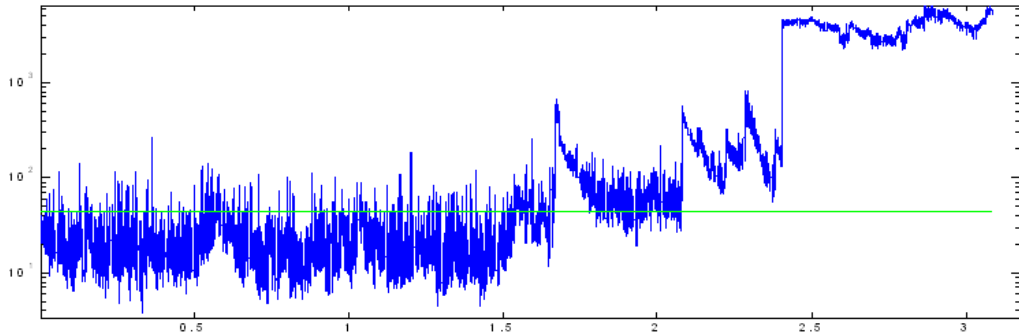


Abb. 7.38: Fehler Mantelriss Antriebstrommel 1: T^2 -Index nach GLR

Wie beim Fehler „Wellenbruch“ fallen wieder die große Empfindlichkeit des T^2 -Index und die hohe Fehlalarmrate von $FAR_{T^2} = 2,93\%$ auf. Zwar hatte der Index im Fehlerfall nicht so extrem große Werte wie bei den Daten des Wellenbruchs, er erreicht jedoch immerhin noch Beträge von 10^4 . Deshalb wurden auch hier die Manipulationen entsprechend der Punkte 1 und 2 im Abschnitt 7.3.1 vorgenommen. Nachfolgende Abbildung 7.39 zeigt die Ergebnisse.

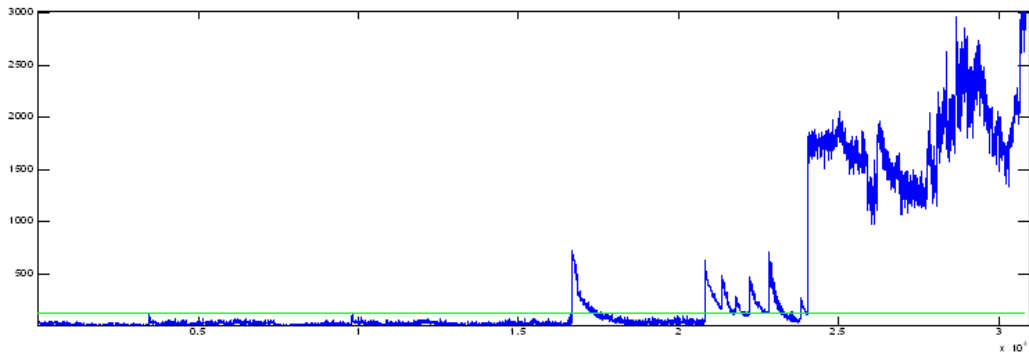


Abb. 7.39: Fehler Mantelriss Antriebstrommel 1: T^2 -Index nach GLR; Schwellwert aus maximalem T^2 -Index der Trainingsdaten und angepasster Kovarianzmatrix

Infolge der Anpassung der Kovarianzmatrix sank die Empfindlichkeit des T^2 -Index sichtbar und der Schwellwert konnte optimal platziert werden. Die Folge ist eine sichere und rechtzeitige Detektion des Fehlers, ähnlich wie bei der PCA. Im Vergleich zur PCA kann man feststellen, dass obwohl bei dem Fehlerdetektionsverfahren nach der Methode des GLR nur ein Index benutzt wurde, durch die vorhandenen Kovarianzen die Fehler-Informationen, die sich bei der PCA im Residuum-Unterraum befinden, nicht verloren gingen und über den T^2 -Index angezeigt wurden.

7.3.3 Sensorfehler

In diesem Abschnitt wurden nun die Daten des Ausfalls des Temperatursensors am linken Lager der Antriebstrommel 1 mit der GLR-Methode untersucht. Das Antrai-

nieren der für die Überwachung notwendigen statistischen Parameter erfolgte wie bei der PCA mit den Daten von Sample 1 bis 6000. In der nachfolgenden Abbildung 7.40 ist wieder der T^2 -Index und der Schwellwert J_{Th} laut Formel 3.72 mit dem Vertrauensbereich von 99,9% zu sehen.

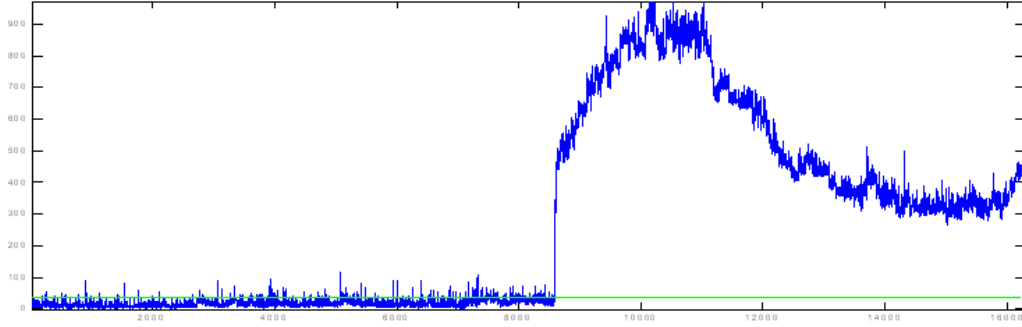


Abb. 7.40: Fehler Sensorausfall: 1; T^2 -Index nach GLR

Im Gegensatz zu den bereits untersuchten Fehlern Wellenbruch und Mantelriss, ist die Dynamik des Indexes diesmal nicht so groß, führt aber auch im fehlerfreien Fall zu vielen Überschreitungen des Schwellwertes und damit zur Anzeige nicht vorhandener Fehler. Da die Fehlalarmrate bei einem Vertrauensbereich von 99,9937% kleiner als 1% war, wurde auf eine Anpassung der Kovarianzmatrix verzichtet und nur der Schwellwert entsprechend Punkt 1 des Abschnittes 7.3.1 neu bestimmt. Die Abbildung 7.41 zeigt die Resultate der on-line-Überwachung.

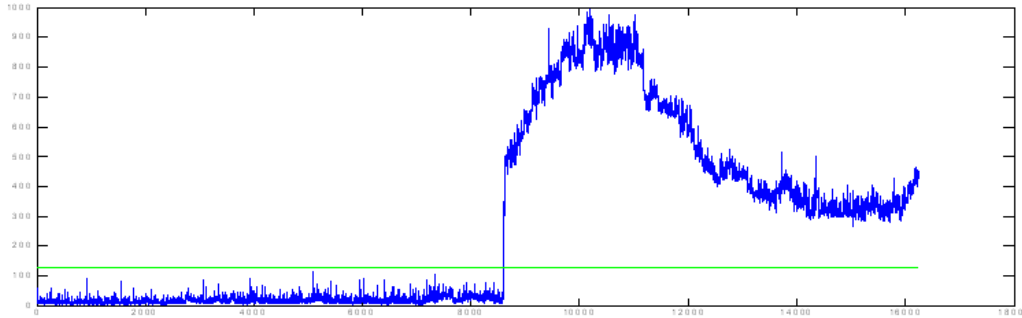


Abb. 7.41: Fehler Sensorausfall: T^2 -Index nach GLR; Schwellwert aus maximalem T^2 -Index der Trainingsdaten und angepasster Kovarianzmatrix

Wie in der Abbildung 7.41 zu sehen, reagiert der Index deutlich auf den Ausfall des Temperatursensors. Während aber die PCA den Sensorfehler nur mithilfe des SPE -Indexes und -Schwellwertes erkennen konnte, zeigte das GLR-Verfahren mit nur einem T^2 -Index den Fehler sicher und rechtzeitig an.

7.3.4 Erhöhte Schwingungen an der Antriebstrommel 2

Abschließend erfolgt die Betrachtung des Fehlerbildes der erhöhten Schwingungen an der Antriebstrommel 2. Das Normalverhalten wurde erneut über die ersten 30000 Sample des verfügbaren Datensatzes angelernt.

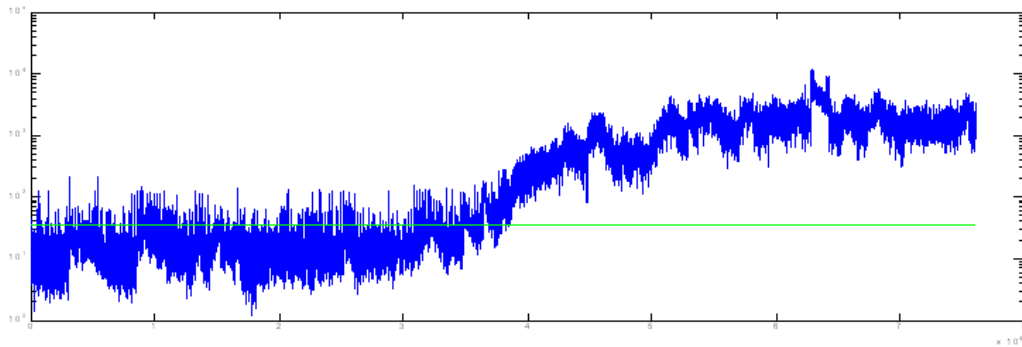


Abb. 7.42: Fehler Erhöhte Schwingungen an Antriebstrommel 2: T^2 -Index nach GLR

Ähnlich wie bei der Auswertung der Fehlerbilder Wellenbruch und Mantelriss mit dem GLR-Fehlerdetektionsverfahren zeigt sich auch bei diesem Fehlerbild (Abbildung 7.42) die große Dynamik des T^2 -Indexes begleitet von einer hohen Fehlalarmrate im fehlerfreien Bereich und riesigen Index-Beträgen von über 10^4 im Fehlerfall. Deshalb wurden anschließend wieder die Veränderungen entsprechend der Punkte 1 und 2 im Abschnitt 7.3.1 vorgenommen, deren Resultate in der nachfolgenden Abbildung 7.43 zu sehen sind.

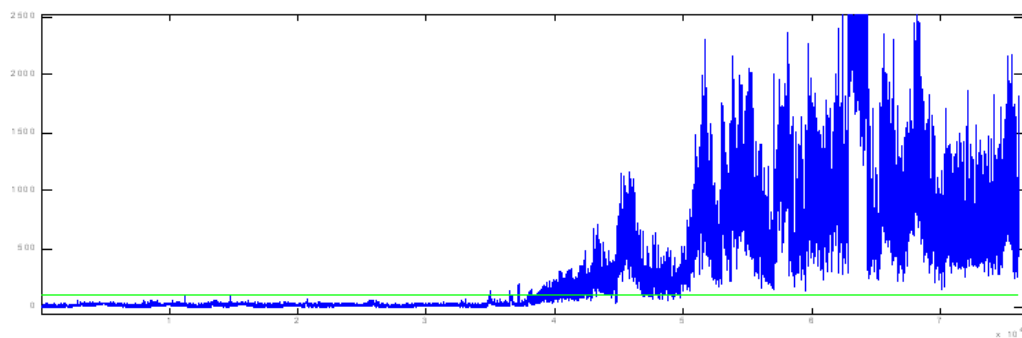


Abb. 7.43: Fehler Erhöhte Schwingungen an Antriebstrommel 2: T^2 -Index nach GLR; Schwellwert aus maximalem T^2 -Index der Trainingsdaten und angepasster Kovarianzmatrix

Das Resultat dieser Untersuchung ist, dass der Schwellwert sicher über den Index-Werten des fehlerfreien Bereiches liegt, jedoch im Fehlerfall die Indizes den Schwellwert deutlich überschreiten. Somit ist das Ergebnis mit dem der PCA vergleichbar.

7.4 Diskussion der Ergebnisse

Im Rahmen dieses Kapitels wurden zwei multivariate, datenbasierte Verfahren zur Fehlerdetektion sowie zwei Verfahren zur Datenvorverarbeitung an einem komplexen mechatronischen System, der Gurtförderanlage CV01, angewendet. Für die Auswertung standen Prozessdaten mit realen, im Abschnitt 6.3 beschriebenen, Fehlerzuständen zur Verfügung. Die für die Anwendung der multivariaten, datenbasierten Verfahren vorausgesetzte Normalverteilung der Daten, war trotz des großen Beobachtungszeitraumes nicht gegeben. Deshalb mussten verschiedene bekannte, aber auch neue Methoden der Datenvorverarbeitung angewendet werden. Letztendlich bleibt jedoch die Frage zu beantworten, ob die multivariaten, datenbasierten Verfahren für die Fehlerdetektion an einem komplexen mechatronischen System praktisch angewendet werden können. Aufschluss sollen die nachfolgend zusammengefassten Ergebnisse der on-line Berechnungen dieses Kapitels bringen.

- Fehlerdetektion mithilfe der PCA

Im Ergebnis der Anwendung der Standard-PCA konnten alle Fehlerzustände rechtzeitig und sicher detektiert werden. In den Fällen des Wellenbruchs und des Mantelrisses an der Antriebstrommel 1 zeigten die Indizes bereits mehrere Tage vor der Entdeckung durch das Personal den Fehlerzustand durch deutliches und dauerhaftes Überschreiten der Schwellwerte an. Die Fehlalarmraten für die fehlerfreien Bereiche sind aber für die praktische Anwendung zu hoch. Die Ursachen dafür wurden im Abschnitt 4.2.2 erklärt. Da die Performanz der Indizes ausreichend groß ist, können durch zusätzliche Maßnahmen, beispielsweise durch die Anpassung der Kovarianzmatrix, wie sie im Abschnitt 4.2.2 beschrieben wurde, bessere Ergebnisse bei Beibehaltung der Bestimmung der Schwellwerte über die F-Statistik erzielt werden.

Während drei von vier Fehlerzustände deutlich durch beide Indizes (T^2 -Index und SPE -Index) angezeigt wurden, erreichte beim Sensorausfall nur der SPE -Index die gewünschte Performanz. Ein Nachteil dieses Verfahrens ist, dass man für die Überwachung zwei Indizes benötigt. Darüber hinaus können durch die teilweise sehr kleinen Eigenwerte im Residuum-Unterraum numerische Probleme bei der Division durch die Eigenwerte während der on-line-Überwachung entstehen.

- Datenvorverarbeitung

Deterministische Anteile in den Erwartungswerten von Messgrößen können zu schlechten Ergebnissen bei der Detektion zufälliger Einflussgrößen mittels der PCA führen. Eliminiert bzw. verkleinert man diese, gewinnen die zufälligen Anteile einen größeren Einfluss. Durch eine lineare Kombination von Messgrößen mit einem starken physikalischen Zusammenhang erzielt man bezüglich der Detektions-Performanz insbesondere im Residuum-Unterraum deutliche Verbesserungen. Zu beachten ist aber, dass man die Differenzbildung, als eine Form der linearen Kombination nicht beliebig durchführen sollte, sondern nur zwischen Messgrößen mit direkten linearen Abhängigkeiten (festen Kopplun-

gen). Andernfalls steigt die Fehlalarmrate im fehlerfreien Bereich deutlich. Die anschließend vorgestellten Ergebnisse zur Anpassung der Kovarianzmatrix zeigen, dass sich diese Methode dann eignet, wenn bereits kleine Instationaritäten zu einer hohen Fehlalarmrate führen.

- Fehlerdetektion mithilfe der Methode des GLR
Die Indizes der fehlerfreien und fehlerbehafteten Bereiche nach der on-line-Überwachung mittels der GLR-Methode weisen große Schwankungen in den Beträgen auf. Das lässt auf eine enorm hohe Empfindlichkeit des T^2 -Indexes gegenüber der Empfindlichkeit der PCA- T^2 -Indizes schließen. Die Ursachen dafür liegen in den im Gegensatz zur PCA noch vorhandenen Kovarianzen. Des Weiteren führt die große Sensibilität des Indexes auch im fehlerfreien Bereich zu deutlichen Überschreitungen des Schwellwertes nach der F-Statistik. Die zum Teil dafür verantwortlichen kleinen Instationaritäten, siehe Abschnitt 4.2.2 kann man trotz verschiedener Methoden der Datenvorverarbeitung nicht vollständig ausblenden. Die zur Bestimmung der Schwellwerte benutzte F-Statistik liefert keinen optimalen Wert. Für eine verbesserte Arbeit mit dem Fehlerdetektionsverfahren nach der GLR-Methode werden zwei Veränderungen vorgeschlagen. Einerseits ist der Schwellwert direkt über den Maximalwert des T^2 -Indexes des Trainingsbereiches zu legen, andererseits kann die Empfindlichkeit des Indexes über eine Anpassung der Kovarianzmatrix, wie in 4.2.2 beschrieben, deutlich reduziert werden. Die Auswertungen der Daten aller vier Fehlerbilder zeigen die deutlich besseren Ergebnisse. Ein wesentlicher Vorteil gegenüber der Fehlerdetektion mit der Standard-PCA-Methode ist, dass nur noch ein Index und ein Schwellwert benötigt werden. Die Bildung von zwei Unterräumen ist nicht notwendig. Trotzdem werden alle Fehler rechtzeitig und sicher angezeigt.

Aufgrund der erzielten Ergebnisse bei der on-line-Überwachung ist es möglich, die beschriebenen Verfahren industriell anzuwenden. Wie bereits beschrieben ist aber auf eine geeignete Datenvorverarbeitung zu achten, um der Forderung nach der Normalverteilung der Daten für die Berechnungen gerecht zu werden. Voraussetzung für den industriellen Einsatz ist die Umsetzung der Verfahren in eine Software. Dafür sind, wie im Abschnitt 5.3 beschrieben, geeignete numerische Softwarebibliotheken auszuwählen. Die Anforderungen an die Rechenleistung eines Überwachungssystems auf der Basis der hier beschriebenen Verfahren sind relativ gering, da es sich insbesondere bei der on-line-Überwachung um einfache Matrizenoperationen handelt.

8 Zusammenfassung und Ausblick

Das Umdenken in der Instandhaltung und das Hinwenden zur Philosophie der Funktionserhaltung und Wertschöpfung führt gleichzeitig zu der Forderung nach anwendbaren und beherrschbaren Fehlerdiagnosesystemen. Datenbasierte Verfahren auf der Grundlage multivariater Methoden eröffnen neue Möglichkeiten bei der Detektion, Lokalisierung und Identifikation von Fehlerzuständen an industriellen Anlagensystemen.

Das in dieser Arbeit verfolgte Ziel ist der Einsatz datenbasierter multivariater Verfahren zur Fehlerdetektion an mechatronischen Systemen. Dahinter steht die Idee, versteckte, für die Überwachung relevante Informationen in einer großen Menge von Datensätzen zu detektieren und nutzbar zu machen, ohne dass der Nutzer sich mit aufwendigen Tätigkeiten zur Datenaufbereitung und Grenzwertermittlung beschäftigen muss. Das Verfahren umfasst Algorithmen zur Merkmalsauswahl, zur Merkmalsreduktion sowie zur Generierung eines „Modells“ im statistischen Sinne, welche mit einer möglichst geringen Anzahl von Parametern eine gute Detektion der Abweichungen vom Normalzustand erlaubt. Es wurde gezeigt, dass die Überwachung und Fehlererkennung in einem komplexen industriellen Prozess auf der Basis multivariater Methoden ohne die Entwicklung eines analytischen Modells der Anlage möglich ist. Die Daten eines Massentransportsystems aus dem Kupferbergbau dienten als Benchmark für Detektierbarkeit von Fehlern mittels der vorgestellten Verfahren. Die Motivation für diese Arbeit bildete die Entwicklung eines Softwareinformationssystems zur Überwachung und Fehlerdiagnose mechatronischer Systeme.

In Kapitel 2 wurde die globale Aufgabenstellung dieser Arbeit vorgestellt und klar gemacht, dass die für die Instandhaltung der Anlagen verantwortlichen Ingenieure und Mechaniker einer großen Datenflut ausgesetzt sind, für deren Auswertung sowohl die Zeit als auch die rechten Mittel fehlen. Daneben existieren jedoch eine Reihe multivariater statistischer Verfahren, die im Bereich der Ökonomie und Biochemie bereits zur Auswertung großer Datenmengen benutzt werden, aber bisher so gut wie keine Anwendung in der Prozessindustrie fanden. Insofern fehlt ein einfaches, unproblematisch anwendbares Softwareüberwachungssystem für die Auswertung großer Datenmengen auf Prozess- und Anlagenfehler. Die Grundlage für ein solches System können datenbasierte Verfahren sein. Die im Rahmen dieser Arbeit untersuchten datenbasierten Methoden und entwickelten Verfahren sollten an einem Benchmark-Prozess eines komplexen mechatronischen Systems verifiziert werden. Davon ausgehend wurden im Abschnitt 2.3 die wesentlichen Merkmale und der grundlegende Aufbau eines mechatronischen Systems vorgestellt. Da mit dieser Arbeit die Grundlagen für ein softwaregestütztes Informationssystem auf der Basis multivariater Methoden gelegt

werden sollen, besteht die Aufgabe darin, einen ersten Entwurf des Softwaredesign vorzustellen.

Die Untersuchung und Erläuterung bereits bekannter multivariater Methoden zur Datenanalyse und zur Fehlerdetektion sowie das Aufzeigen der wesentlichen mathematischen und statistischen Zusammenhänge und Einflussgrößen ist das Ziel des 3. Kapitels. Als Voraussetzung für eine erfolgreiche Detektion von Anlagenzustandsfehlern benötigt man für die Analyse geeignete Prozessdaten. Deshalb werden in dem Abschnitt 3.2 bekannte Probleme mit der Prozessdatenbasis und Methoden der Datenaufbereitung aufgezeigt.

Der Abschnitt 3.3 beschäftigt sich mit grundlegenden datenbasierten Methoden zur Fehlererkennung. Den Schwerpunkt bilden die Fehlerdetektionsverfahren auf der Basis der PCA- und der GLR-Methode. Dabei wird auf die Integration dieser Methoden in komplette Verfahren zur Fehlerdetektion mit den beiden, immer wiederkehrenden Schritten „off-line-Training“ und „on-line-Überwachung“ eingegangen. Da industrielle Prozesse ein sehr unterschiedliches Verhalten haben, in deren Folge die klassischen Methoden der PCA und GLR nicht immer zuverlässige Ergebnisse liefern, wurden leicht abgewandelte Verfahren der Fehlerdetektion mittels PCA und GLR vorgestellt. Dazu gehören die dynamische PCA, für Prozesse bei denen der aktuelle Status des Systems von Werten aus der Vergangenheit abhängt oder auch die rekursive PCA für zeitvariante Prozesse.

Das Ziel der Entwicklung von datenbasierten Verfahren zur Fehlererkennung ist die Einbettung dieser in ein Softwareinformationssystem, dessen grundlegendes Design im Rahmen dieser Arbeit vorgestellt werden soll. Damit im Ergebnis des Entwicklungsprozesses eine sicher funktionierende, intuitiv bedienbare, wartbare sowie erweiterbare Software entsteht, muss gerade auf den Designentwurf großes Augenmerk gelegt werden. Die Techniken und Herangehensweisen beim Softwaredesign haben in den letzten Jahren einen gewaltigen Entwicklungsschub erfahren. Die wesentlichen Aspekte, Kriterien und Richtlinien, die beim Design einer modernen Software zu beachten sind, erläutert der Abschnitt 3.4.

Der letzte Abschnitt von Kapitel 3 befasst sich mit dem Aufbau, dem Prozessverhalten und charakteristischen Fehlerzuständen einer Gurtförderanlage. Es bildet die Grundlage für das Verständnis des Verhaltens der später untersuchten Prozessdatenreihen. Die Beschaffenheit der Anlage entspricht einem typischen mechatronischen System, dessen wesentliche Merkmale im Abschnitt 2.3 erläutert wurden.

In Kapitel 4 werden Anpassungen und Erweiterungen der Algorithmen vorgestellt, die helfen sollen, trotz vorhandener kleiner Instationaritäten in und Korrelationen zwischen den Messdaten eine optimale Fehlerdetektion durchzuführen. Der Abschnitt 4.2.1 widmet sich insbesondere störenden Dynamiken im Erwartungswert, welche bei vorhandenen Redundanzen durch die gezielte lineare Kombination von Messgrößen vermindert werden können. Es konnte gezeigt werden, dass sich die Detektierbarkeit multiplikativer und additiver Fehler mithilfe der PCA durch die lineare Kombination von Messgrößen verbessert.

Die multivariate Normalverteilung der Daten ist eine wesentliche Voraussetzung für

die effektive Funktion der beschriebenen datenbasierten Verfahren. Wie im Abschnitt 4.2.2 beschrieben, kann man nicht in jedem Fall erfolgreich die stationären Bereiche von den instationären trennen, was zulasten der Normalverteilung der Prozessdaten geht. Deshalb wurde zur Erhöhung der Robustheit des Verfahrens die Anwendung einer Kovarianzmatrix auf der Basis künstlicher, normalverteilter Daten vorgeschlagen. Das Generieren der Modelldaten erfolgt mithilfe der Erwartungswerte und Standardabweichungen der Trainingsdaten.

Die Ausführungen im Abschnitt 4.3 zeigen, dass die Methode des Generalized-Likelihood-Quotienten, basierend auf einem Maximum-Likelihood-Schätzer für den Erwartungswert zur gleichen Teststatistik führt, welche bei der PCA für die Hauptkomponenten zur Anwendung kommt. Eine Aufteilung in zwei Unterräume, wie bei der PCA sowie der Schritt der Singulärwertzerlegung ist somit überflüssig. In letzter Konsequenz bedeutet es die Möglichkeit der Anwendung der GLR- anstelle der PCA-Technik.

Zur Überwachung von zeitinvarianten Prozessen kann man die RPCA einsetzen. Um das Gleiche mit der Methode des GLR zu erreichen, wurden drei adaptive GLR-Verfahren entwickelt und beschrieben. Der Einsatz der adaptiven Methoden ist aber nicht unproblematisch, da auch Fehlerzustände adaptiert werden können, was letztendlich eine erfolgreiche Detektion verhindert.

Das Kapitel 5 beschreibt den Designentwurf des Softwareinformationssystems. Dieser basiert auf der Entwicklung einer modernen verteilten Architektur mit einem Web-Client. Im Design kommen bewährte Design-Patterns und Frameworks zum Einsatz, womit die Forderungen nach guter Wartbarkeit und Erweiterbarkeit erfüllt werden. Der Entwurf enthält bereits erste Ansätze für das konkrete Klassendesign. Diese sind jedoch während der weiteren Entwicklung des Systems zu vervollständigen und zu verfeinern.

Der Anpassung und Erweiterung der vorgestellten Verfahren zur Fehlerdetektion folgt in den Kapiteln 6 und 7 die Anwendung an dem konkreten Benchmark-Prozess. In Kapitel 6 wird zunächst der Aufbau der Gurtförderanlage CV01 mit ihren wesentlichen Parametern beschrieben. Diesem Abschnitt schließt sich die Schilderung des Aufbaus und Inhaltes der Prozessdatensätze sowie der zur Verfügung stehenden Sensorik und zugehörigen Abstraten an. Es folgt eine Erläuterung der Verfahren zur Datenvorverarbeitung, die an den Prozessdaten angewendet wurden. Der Abschnitt 6.3 enthält eine Beschreibung der Fehlerbilder, zu denen Prozessdaten zur Verfügung standen. Der Zustand der Prozessdaten sowie die darin enthaltenen Fehlerzustände stellen eine Herausforderung an die Leistungsfähigkeit der Verfahren zur Datenvorverarbeitung und Fehlerdetektion dar.

Das Kapitel 7 beschreibt nun die konkreten Ergebnisse der Anwendung der vorgestellten und entwickelten Verfahren an den vorhandenen Prozessdaten mit den fehlerfreien und fehlerbehafteten Zuständen. Zum Einsatz kamen zwei multivariate, datenbasierte Verfahren zur Fehlerdetektion sowie zwei Verfahren zur Datenvorverarbeitung. Die dargestellten Analysen zeigen, dass sich die Algorithmen der Hauptkomponentenanalyse für die Überwachung eines industriellen Massentransportprozesses verwenden lassen. Als besonders leistungsfähig hat sich das Verfahren zur Fehlerde-

tektion nach der Methode des Generalized-Likelihood-Ratio erwiesen, wobei die enorme Empfindlichkeit des für die Überwachung genutzten T^2 -Indexes bei Verwendung der Kovarianzmatrix zu einer großen Fehlalarmrate führen kann. Alle Fehler konnten sowohl durch das Fehlerdetektionsverfahren auf der Basis der PCA-Methode als auch das auf der Basis der GLR-Methode eindeutig und rechtzeitig erkannt werden. Für den Wellenbruch und den Mantelriss gelang das bereits Tage vor dem eigentlichen Entdecken durch das Bedien- und Wartungspersonal. Die hohen Fehlalarmraten sind sowohl für die PCA- als auch GLR-Methode ein Problem. Deshalb wurden im Abschnitt 7.2 zwei Verfahren zur Datenaufbereitung getestet. Mit ihnen kann einerseits die Empfindlichkeit der Überwachung optimiert werden, sodass kleine Instationaritäten keine so großen Auswirkungen auf den Index haben und andererseits können deterministische Anteile im Erwartungswert ausgeblendet werden. Beide Verfahren führen zu den gewünschten Ergebnissen. Vergleicht man das Fehlerdetektionsverfahren auf Basis der PCA mit dem auf Basis des GLR, so kann man feststellen, dass das Standard-GLR-Verfahren eine so hohe Empfindlichkeit hat, dass sich praktisch einsetzbare Schwellwerte nicht bilden lassen. Das führt in letzter Konsequenz dazu, dass es zur Fehlerdetektion an diesem Prozess ohne ergänzende Verfahren zur Schwellwertbildung und zur Verringerung der Empfindlichkeit nicht verwendbar ist. Das GLR-Verfahren hat jedoch den Vorteil, dass man nur einen Index und einen Schwellwert für die Überwachung benötigt, ohne Einbußen in der Performance zu haben. Dagegen ist das Standard-PCA-Verfahren deutlich weniger empfindlich und die über die F-Verteilung ermittelten Schwellwerte zeigen zusammen mit den Indizes eine gute Performance. Von Nachteil ist, dass zur Analyse zwei Unterräume und zu jedem Unterraum ein separater Index und Schwellwert gebildet werden muss. Hinzu kommt noch die rechentechnisch aufwendigere Singulärwertzerlegung, welche bei dem GLR-Verfahren nicht notwendig ist. Bei entsprechend guter Datenvorverarbeitung erweisen sich beide Verfahren als geeignet für den praktischen Einsatz, obwohl man dem GLR-Verfahren, bei entsprechender Anpassung der Kovarianzmatrix, wegen seiner Einfachheit bezüglich der Berechnung und Auswertung den Vorzug geben kann.

Zusammenfassend kann man feststellen, dass es in den Verfahren bezüglich der Bestimmung eines einzigen Indexes und eines Schwellwertes sowie der Auswertelogik Optimierungspotenziale gibt. Für die PCA-Verfahren finden sich in den Dokumenten [52] und [17] Vorschläge zur Bildung eines einzigen gemeinsamen Indexes und Schwellwertes für die Unterräume der Hauptkomponenten und der Residuen, welche durch weiterführende Untersuchungen auf ihre Eignung für die Fehlerdetektion speziell an Gurtförderanlagen und allgemein an mechatronischen Systemen zu testen sind. Wie die Ergebnisse dieser Arbeit zeigen, kann die multivariate Normalverteilung der Daten nicht immer vorausgesetzt werden. Dadurch kommt es vor, dass schon im fehlerfreien Bereich die Schwellwerte immer wieder kurzzeitig überschritten werden. Da die Überschreitungen im fehlerfreien Zustand keinen permanenten Charakter haben, könnte die Fehlalarmrate u.a. auch durch Veränderung der Auswertelogik gesenkt werden. Ein Ansatz ist die Tiefpassfilterung und damit Glättung des Indexes, was aber zu einer leicht verzögerten Erkennung des Fehlers führt. Ein anderer ist

die Zählung der kurzzeitigen Überschreitungen, wobei erst nach einer bestimmten Anzahl von Überschreitungen in einem festgelegten Zeitbereich ein Alarm ausgelöst wird.

Wie man dem Abschnitt 2.2 und der darin enthaltenen Abbildung 2.1 entnehmen kann, gehören zu einem Prozessüberwachungssystem auch die Lokalisierung und die genaue Identifikation von Fehlern. Diese beiden Bereiche wurden in dieser Arbeit nicht untersucht. Im Rahmen der Entwicklung des Softwareinformationssystems zur Anlagenüberwachung lassen sich folgende Aufgaben für weiterführende Ansätze auf dem Gebiet der datenbasierten Fehlerdiagnose formulieren:

- Lokalisierung mit Reconstruction Based Sensor Fault Detection, Reconstruction Based Contributions nach [51] und [44]
- Identifizierung mit Fisher Discriminant Analysis nach [58] und Reconstruction Based Fault Identification nach [51] und [52].

Mithilfe dieser und weiterer datenbasierter Verfahren zur Lokalisierung und Identifikation ist man dann auch in der Lage, die Verursacher von Fehlern zu ermitteln. Ein so ausgebautes datenbasiertes Softwareinformationssystem unterstützt die Instandhaltung bei der Ermittlung von Schwachstellen und Kostentreibern.

Anhang A

A.1 Funktionelle Abläufe

Funktioneller Gesamtablauf

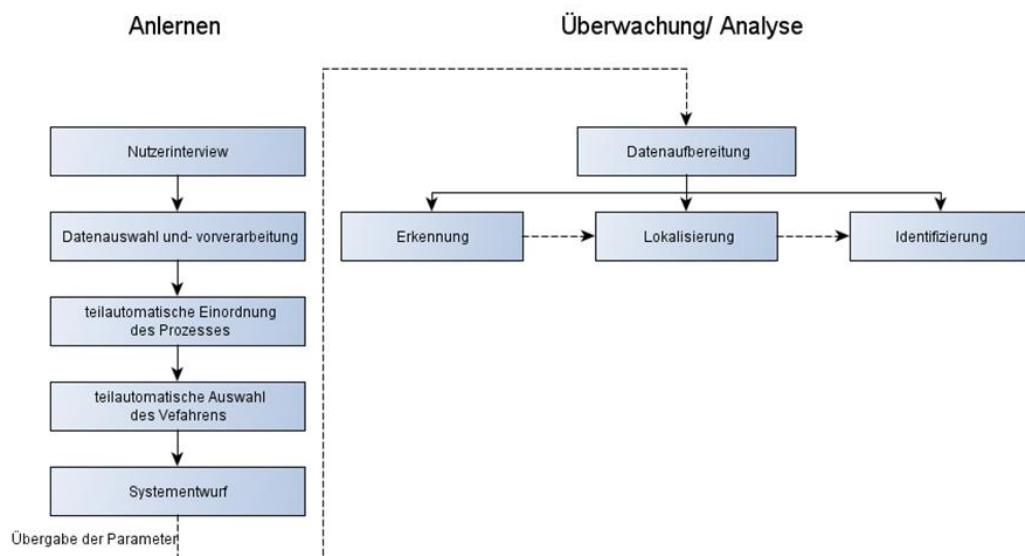


Abb. A.1: Funktioneller Ablauf der Software aus [21]

Charakterisierung des Prozesses

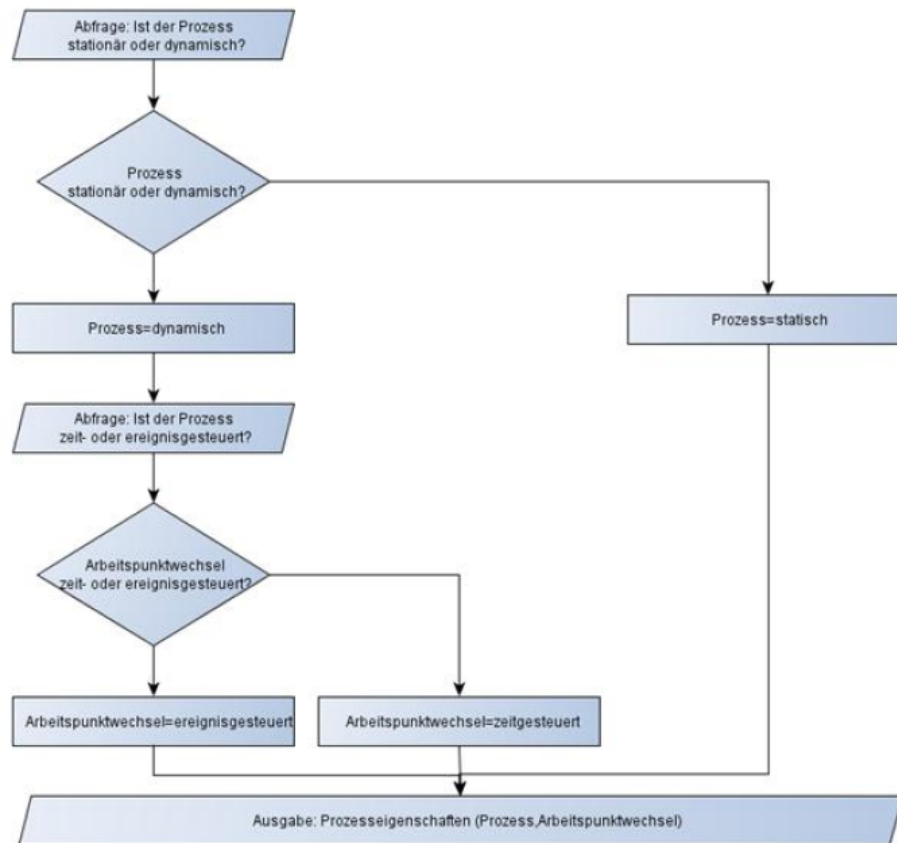


Abb. A.2: Funktioneller Ablauf zur Charakterisierung des Prozesses aus [21]

Datenvorverarbeitung

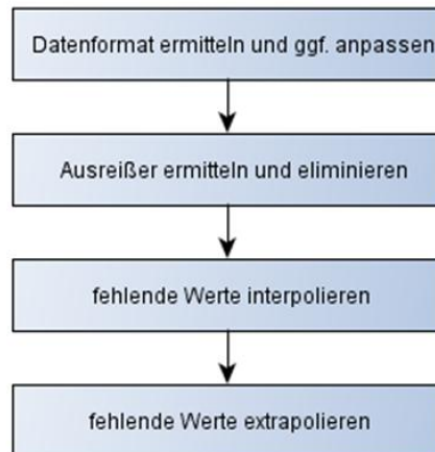


Abb. A.3: Funktioneller Ablauf Datenvorverarbeitung aus [21]

Anhang B

Das Klassendesign ausgewählter Pakete

B.1 Das Paket Data-Source

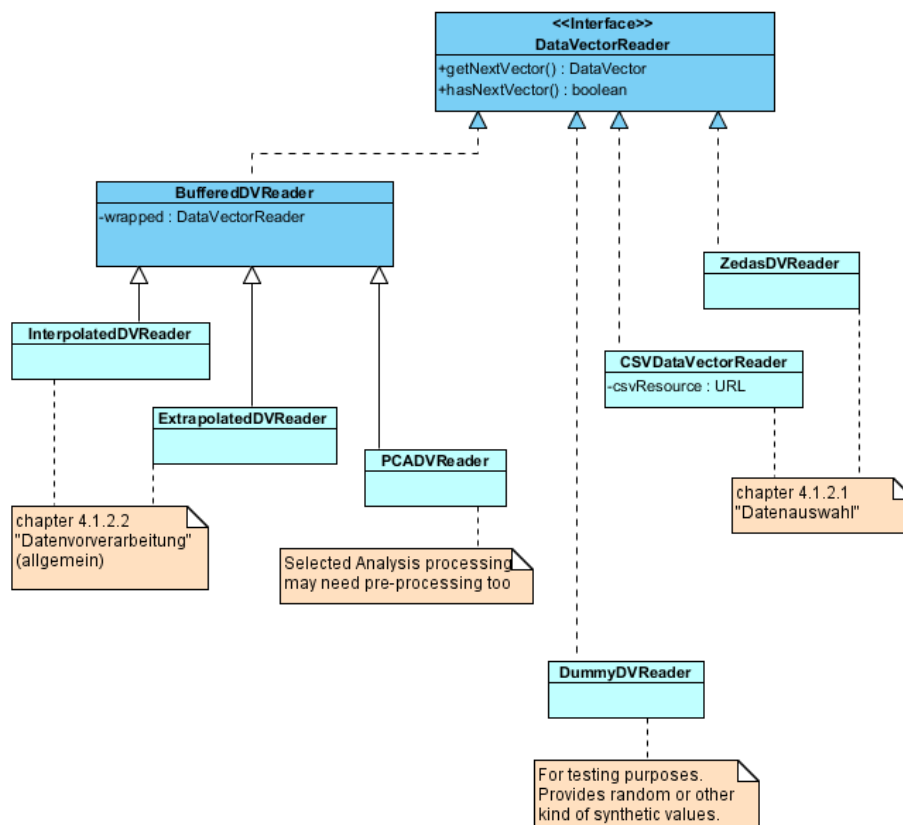


Abb. B.1: Klassendesign Data-Source aus [21]

Klasse DataVectorReader (Interface)

Basisinterface aller Reader-Klassen

Methoden

- `read()`
Liest den nächsten Datenvektor aus und gibt ihn zurück. Ist kein Vektor verfügbar, blockiert die Methode solange, bis ein Wert zurückgeliefert werden kann. Es wird null zurückgeliefert, wenn das Ende des Datenstroms erreicht ist. Wird die Methode anschließend erneut aufgerufen, blockiert diese dauerhaft.
- `ready()`
Liefert true, wenn garantiert wird, dass der nächste Aufruf von `read()` nicht blockiert.

Klasse `BufferedDVReader`

Bietet einen Zwischenspeicher (Puffer) für das Lesen von Vektoren aus einem anderen `DataVectorReader` an. Der Puffer ist bestrebt, immer gefüllt zu bleiben und enthält nur Datenvektoren (null ist nicht erlaubt). Ein leerer Puffer ist nicht gleichbedeutend mit dem Ende des Datenstroms.

Methoden

- `read()`
Liest den nächsten Wert aus dem Puffer aus. Zugleich wird versucht, den Puffer wieder vollständig aufzufüllen, jedoch nur soweit, wie die interne `DataVectorReader`-Quelle nicht blockiert. Ansonsten besitzt er dasselbe Verhalten wie der `DataVectorReader`.
- `ready()`
siehe `DataVectorReader`

B.2 Das Paket Processing

In diesem Paket werden die Arbeitsschritte/ Analyse eines einzelnen DatenVektors spezifiziert. Ein kompletter Workflow bzw. die Abarbeitung mehrerer Vektoren ist im Paket Job geklärt.

Klasse `AnalysisConfig` (Interface)

Diese Klasse enthält sämtliche Konfigurationsparameter, die für die konkrete Analyse benötigt werden. Die Parameter müssen bereits ermittelt worden (automatisch oder manuell) und validiert sein. Es besteht die Möglichkeit, die Analyse-Strategie zur Laufzeit auszutauschen.

Methoden

- `getAnalysisStrategy(): AnalysisStrategy`
Liefert eine Strategie, welche die `AnalysisConfig` interpretieren und damit im Zusammenspiel mit einem `DataVektor` ein `AnalysisResult` erstellen kann. Es muss immer eine `AnalysisStrategy` gesetzt sein (null ist nicht erlaubt)

Klasse `AnalysisStrategy` (Interface)

Diese Klasse führt die eigentliche Analyse (im mathematische Sinne) durch, ohne jedoch das Ergebnis zu bewerten oder Handlungen einzuleiten.

Methoden

- `analyze (DataVector, AnalysisConfig): AnalysisResult`
Sie ist die einzige zu implementierende Methode welche den übergebenen `DataVector` analysiert. Ergebnis ist ein `AnalysisResult`-Objekt. Für die Analyse wird i.d.R. eine Anzahl an Parametern benötigt, welche in der `AnalysisConfig` enthalten sind. Der zu übergebene `DataVector` ist niemals null. Ebenso darf `AnalysisResult` niemals null sein

Klasse `AnalysisResult` (Interface)

Die Klasse liefert das konkrete Ergebnis einer Analyse. Die Minimalimplementierung muss mindestens `AnalysisConfig` und `DataVector` kennen, welche für dieses Ergebnis verantwortlich sind. Eine konkret implementierte `AnalysisStrategy` (z.B. `PCAAnalysis`) wird aber in der Regel ein dazu passendes, entsprechend abgeleitetes `AnalysisResult`-Objekt produzieren, welches wesentlich mehr Informationen (analysespezifisch) liefert.

Methoden

- `getDataVector(): DataVector`
Liefert den Datenvektor, der zu diesem Ergebnis geführt hat.
- `getConfig(): AnalysisConfig`
Gibt die analysespezifischen Konfigurationsparameter, die zur Berechnung verwendet wurden, zurück.
- `getTimestamp(): long`
Liefert den Zeitpunkt der Ergebnisferigstellung.

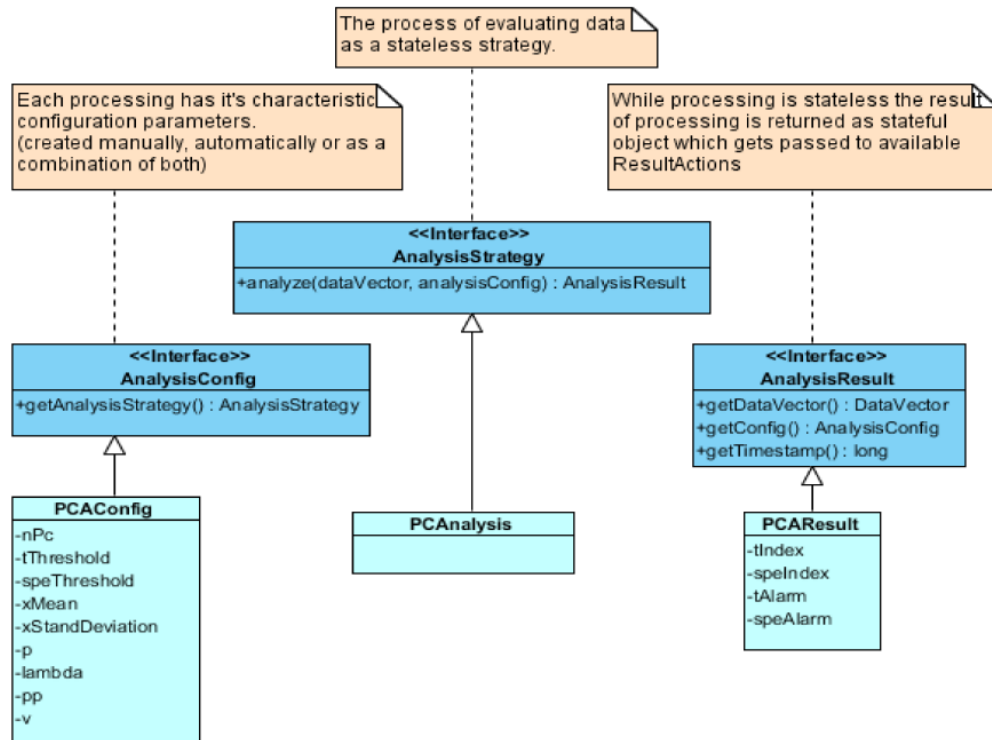


Abb. B.2: Klassendesign für das Paket Processing aus [21]

B.3 Das Paket Action

Das Paket Action behandelt stets das Ergebnis der letzten Analyse.

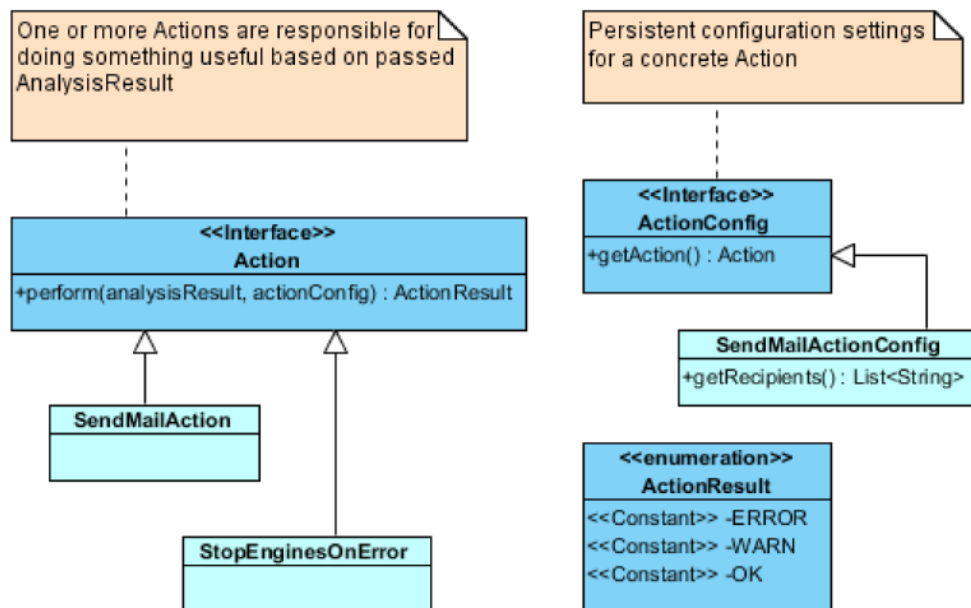


Abb. B.3: Klassendesign Ereignisbehandlung im Paket Action [21]

Klasse ActionConfig (Interface)

Diese Klasse funktioniert analog zur AnalysisConfig, muss aber mindestens die auszuführende Action bereitstellen. Sie wird in konkreten Ableitungen wesentlich mehr Parameter enthalten (z.B. eine SendMailActionConfig, welche eine Email-Empfängerliste verwaltet und als Action eine SendMailAction zurückgibt).

Methoden

- `getAction(): Action`

Klasse Action (Interface)

In dieser Klasse entscheidet man anhand eines vorliegenden AnalysisResult, ob Handlungsbedarf besteht und leitet diesen auch ein. Die übergebene ActionConfig liefert alle dafür notwendigen Zusatzparameter. (z.B. eine SendEmailAction welche Mails versendet und die Empfänger der SendMailActionConfig entnimmt)

Methoden

- `perform(AnalysisResult, ActionConfig): ActionResult`
Einzige zu implementierende Methode. Ähnlich der AnalysisConfig, welche analysespezifische Parameter bereitstellt, ist für die Action die ActionConfig vorgesehen. Die Klasse ActionConfig ist nicht generisch und muss entsprechend gecasted werden.

Klasse ActionResult (Enumeration)

Das Ergebnis einer ausgeführten Action. Anmerkung: Momentan (der Einfachheit halber) eine Aufzählung, kann sich jedoch noch ändern und ähnlich dem AnalysisResult ein aussagekräftigeres Objekt werden.

Methoden

- OK, WARN, ERROR
Nicht das Ausführungsergebnis der Action an sich, sondern wie die Action eine AnalysisResult interpretiert hat. (z.B. ERROR heißt, dass die Action Fehlermaßnahmen ergriffen hat, aber nicht selbst ein Fehler produziert hat)

B.4 Das Paket Job

Dieses Paket enthält komplette Workflows in Form von Jobs und übernimmt das Steuern und Überwachen der Jobs.

1. Anlegen und Steuern von Jobs

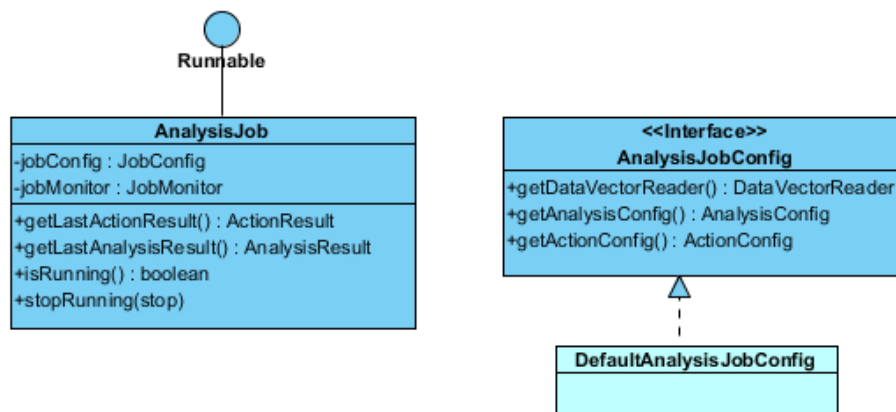


Abb. B.4: Klassendesign Job aus [21]

Klasse AnalysisJobConfig (Interface)

Liefert alle Parameter die notwendig sind, einen Job zu starten. Dazu gehört die Datenquelle, welche die Datenvektoren liefert, die für die Vektoren notwendigen Analyse Parameter und die für die Ereignisbehandlung nötigen Action-Parameter.

Methoden

- `getDataVectorReader(): DataVectorReader`
Liefert die Vektoren welche analysiert werden sollen
- `getAnalysisConfig(): AnalysisConfig`
Definiert die Analyse
- `getActionConfig(): ActionConfig`
Definiert die Ereignisbehandlung

Klasse **AnalysisJob**

Diese Klasse führt eine komplette Analyse durch, so wie sie durch eine `AnalysisJobConfig` vorgegeben ist. Da dies ein rechen- und zeitaufwendiger Prozess sein kann, sollte dieser Job in einem separaten Thread laufen. Das ist jedoch keine Pflicht.

Methoden

- `getLastActionResult(): ActionResult`
Liefert das Ergebnis der letzten Ereignisbehandlung
- `getLastAnalysisResult(): AnalysisResult`
Liefert das Ergebnis der letzten Analyse
- `getJobMonitor(): AnalysisMonitor`
Liefert einen Überwachungsmonitor
- `isRunning(): boolean`
True wenn der Job noch läuft
- `stopRunning(boolean)`
Veranlasst den Job sobald wie möglich zu stoppen

Der `AnalysisJob` läuft in einer Dauerschleife die erst beendet wird, wenn keine Vektoren mehr verfügbar sind oder der Nutzer manuell abgebrochen hat.

Jeder Schleifenzyklus beinhaltet:

- `DataVectorReader`, `AnalysisConfig` und `ActionConfig` auslesen,
- nächsten `DatenVektor` holen (sofern verfügbar),
- an konfigurierte `AnalysisStrategy` weiterreichen,
- erzeugtes `AnalysisResult` an konfigurierte Action weiterreichen.

2. Überwachen von Jobs

Klasse Monitor (Interface)

Methoden

- `getMonitorable(): Object`
Liefert das Objekt, welches überwacht wird
- `getLastEvents(): List<MonitorEvent>`
Liefert eine Liste gesammelter `MonitorEvents`. Die Liste ist begrenzt und die ältesten Events werden stets verworfen

B.5 Das Paket Context

Ein Kontext oder Scope definiert einen Lebenszyklus oder auch „Haltbarkeit“ eines Objektes. Ein implizites Objekt wird automatisch von der ausführenden Umgebung bereitgestellt und dabei mit einem solchen Kontext versehen.

1. Vorhandene Kontexttypen

- **Application-Scope:**
Applikationsweite Objekte entsprechen dem Lebenszyklus der Java-VM. Einmal instanziiert werden Objekte mit diesem Scope erst wieder bei Beenden der Applikation zerstört. (Analogie zum Web: `ServletContext`; oder allgemein: Singleton)
- **Session-Scope:**
Sessionbasierte Objekte sind an einen Nutzer(Client) gebunden. (Analogie zum Web: `HttpSession`)
- **Request-Scope:**
Requestbasierte Objekte sind an eine Nutzeraktion gebunden. (Analogie zum Web: `HttpServletRequest`)

2. Vorhandene implizite Objekte

Alle impliziten Objekte sind erreichbar durch die statische Methode `MVRequest.getCurrentRequest` welche den aktuellen Request liefert.

Klasse MVApplication (Scope: Application)

Es existiert genau eine Instanz dieser Klasse innerhalb der gesamten Applikation. In dieser Klasse werden Jobs gestartet sowie überwacht und hier finden sich Services und Hilfsmethoden/-klassen, die dem Singleton-Muster entsprechen (demzufolge Thread-safe sind).

Methoden

- `getMathService(): MathService`
Liefert den zu verwendenden MathService
- `getRunningJobs(): List<AnalysisJob>`
Liefert alle derzeit laufenden Jobs
- `startNewJob(job)`
Startet einen neuen Job
- `getAttributes(): Map`
Liefert eine zur freien Verfügung stehende Attribut-Map. Objekte, die einen Application-Scope erhalten sollen, können hier jederzeit abgelegt/entfernt werden.

Klasse MVSession (Scope: Session)

Es existiert genau eine Instanz dieser Klasse pro User/Client. Der User/Client kann sich logisch gesehen ändern, obwohl er dies physisch nicht getan hat, z.B. bei einem Wechsel von einem anonymen Nutzer zu einem bekannten Nutzer. In diesem Fall wird auch die alte MVSession verworfen und entsprechend neu aufgebaut.

Methoden

- `getApplication(): MVApplication`
Liefert die MVApplication. Alle vorhandenen MVSession-Instanzen geben stets dieselbe MVApplication-Instanz zurück.
- `getAttributes(): Map`
Liefert eine zur freien Verfügung stehende Attribut-Map. Objekte, die einen Session-Scope erhalten sollen, können hier jederzeit abgelegt bzw. entfernt werden.

Klasse MVRequest (Scope: Request)

Es existiert genau eine Instanz dieser Klasse pro Nutzerinteraktion (z.B. durch das Betätigen eines Buttons inklusive des Abarbeitens der daraus folgenden Logik. Danach ist der Request beendet)

Methoden

- `getSession(): MVSession`
Liefert die zu diesem Request gehörende `MVSession`. Mehrere Requests von ein und demselben Nutzer/Client liefern stets dieselbe `MVSession`-Instanz.
- `getAttributes(): Map`
Eine zur freien Verfügung stehende Attribut-Map - Objekte, die einen Request-Scope erhalten sollen, können hier jederzeit abgelegt/entfernt werden.

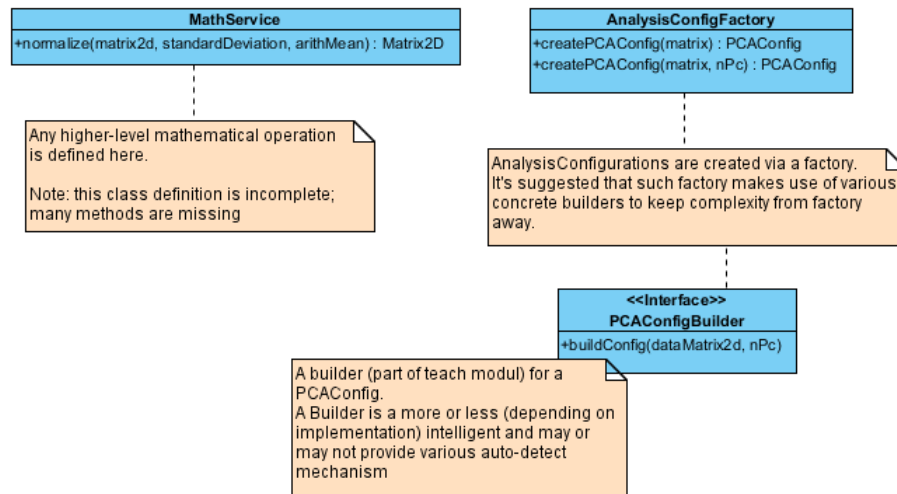


Abb. B.5: Klassendesign Math-Service aus [21]

Literaturverzeichnis

- [1] ARBEITSKREIS, Namur: Asset Management. In: *ATP 11.2007* (2007)
- [2] BASSEVILLE, M. ; NIKIFOROV, I.V.: *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, 1993
- [3] BÖGE, A.: *Handbuch Maschinenbau*. GWV Fachverlage GmbH, Wiesbaden : Vieweg & Sohn Verlag, 2007
- [4] BOSCH, J.: *Design and use of software architectures*. Addison-Wesley, 2000
- [5] BOURIER, G.: *Wahrscheinlichkeitsrechnung und schließende Statistik*. Gabler - GWV Fachverlage GmbH, Wiesbaden, 2009
- [6] BOX, G. E. P.: Some theorems on quadratic forms applied in the study of analysis of variance problems, I. effect of inequality of variance in the one-way classification. In: *Ann. Math. Statistics* (1954)
- [7] CHIANG, L. H. ; RUSSEL, E. L. ; BRAATZ, R. D.: *Fault Detection and Diagnosis in Industrial Systems*. London : Springer-Verlag, 2001
- [8] CHOI, S. W. ; MARTIN, E. B. ; MORRIS, A. J. ; LEE, I.: Adaptive Multivariate Statistical Process Control for Monitoring Time-varying Processes. In: *Industrial & Engineering Chemistry Research* (2006)
- [9] CLEMENTS, P. ; BACHMAN, F. ; BASS, L. ; GARLAN, D. ; IVERS, J. ; LITTLE, R. ; MERSON, P. ; NORD, R. ; STAFFORD, J.: *Documenting Software Architectures, Views and Beyond*. Addison-Wesley, 2010
- [10] COPELAND, T.: *PMD Applied*. Centennial Books, 2005
- [11] CORPORATION, Oracle: *Java Platform, Enterprise Edition (Java EE) Technical Documentation*. <http://download.oracle.com/javaee/>, 2009/2010. – [Online accessed 2011-02-24]
- [12] CORPORATION, Oracle: *JavaServer Faces Technology - Documentation*. <http://www.oracle.com/technetwork/java/javaee/documentation/index-137726.html>, 2010. – [Online accessed 2011-02-24]
- [13] DASZYKOWSKI, M. ; KACZMAREK, K. ; HEYDEN, Y. ; WALCZAK, B.: Robust statistics in data analysis-A review (Basic concepts). In: *Chemometrics and Intelligent Laboratory Systems* Vol. 85 (2007), Nr. 203-219

- [14] DING, S. X.: *Model-based Fault Diagnosis Techniques*. Berlin Springer Verlag, 2008
- [15] DING, S. X. ; JEINSCH, T. ; ENGEL, P. ; JAHN, W.: Verfahren zur Offline- und /oder Onlineüberwachung und Grundlage zur Merkmalgenerierung einer Optimierung sowie Bearbeitung des Zustands einer Gurtförderanlage. In: *Amt. Kennz. 199 08 317.7* (1999)
- [16] DING, S. X. ; ZHANG, P. ; DING, E. ; YIN, S. ; NAIK, A. ; DENG, P. ; GUI, W.: *On the application of PCA technique to fault diagnosis*. Tsinghua Science and Technology, 2010
- [17] DING, S. X. ; ZHANG, P. ; JEINSCH, T. ; DING, E. L. ; ENGEL, P. ; GUI, W.: A survey of the application of basic data-driven and model-based methods in process monitoring and fault diagnosis. In: *18th IFAC World Congress, (IFAC WC 2011)* (2011)
- [18] DIVISION:, Mathematical & Computational S.: *Java Numerics*. <http://math.nist.gov/javanumerics>, 2010-03-03. – [Online accessed 2010-11-12]
- [19] ELSHENAWY, L. M. ; YIN, S. ; NAIK, A. S. ; DING, S. X.: Efficient Recursive Principal Component Analysis Algorithms for Process Monitoring. In: *Ind. Eng. Chem. Res.* (2009)
- [20] ENERGIESYSTEMTECHNIK (IWES), Fraunhofer I. u.: Windenergie Report Deutschland 2010. In: *erstellt im Rahmen des Forschungsvorhabens „Monitoring der offshore-Windenergienutzung-Offshore-WMEP “* (2010)
- [21] ENGEL, P. ; GRUSS, G. ; STRIETZBAUM, D.: *Functional Design Specification zeda\$@asset*. 2011
- [22] FAHRMEIR, L. ; HAMERLE, A. ; (HRSG.), G. T.: *Multivariate statistische Verfahren*. 2., erweiterte Auflage. Berlin : Walter de Gruyter & Co. Verlag, 1996
- [23] FOWLER, M.: *Analysis Patterns: Reusable Object Models*. Addison-Wesley, 1997
- [24] GENSCHEL, U. ; BECKER, C.: *Schließende Statistik Grundlegende Methoden*. Springer-Verlag Berlin Heidelberg, 2005
- [25] GEORGII, H.: *Stochastik - Einführung in die Wahrscheinlichkeitstheorie der Statistik*. Berlin, New York : Walter de Gruyter-Verlag, 2007
- [26] HANDL, A.: *Multivariate Verfahren Theorie und Praxis unter besonderer Berücksichtigung von S-Plus*. New York, Berlin : Springer-Verlag, 2002
- [27] HANDL, A.: *Multivariate Analysemethoden*. New York, Berlin : Springer-Verlag, 2007
- [28] HE, X. B. ; YANG, Y. P.: Variable MWPCA for Adaptive Process Monitoring Processes. In: *Industrial & Engineering Chemistry Research* (2008)

- [29] HOTELLING, H.: Multivariate quality control illustrated by the air testing of sample bombsights. In: *In C. Eisenhart, M.W. Hastay, and W.A. Wallis(Eds.), Techniques of Statistical Analysis, pp. 113-184. McGraw-Hill, New York (1947)*
- [30] ISERMANN, R.: *Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance.* Berlin, Heidelberg : Springer-Verlag, 2006
- [31] ISERMANN, R.: *Mechatronische Systeme: Grundlagen.* Berlin, Heidelberg : Springer-Verlag, 2007
- [32] JACKSON, J. E. ; MUDHOLKAR, G. S.: Control procedures for residuals associated with principal component analysis. In: *Technometrics* (1979)
- [33] JANKER, Ch. G.: *Multivariate Lieferantenbewertung: Empirisch gestützte Konzeption eines anforderungsgerechten Bewertungssystems.* Wiesbaden : Deutscher Universitäts-Verlag/GWV Fachverlage GmbH, 2004
- [34] JANSEN, M.: Entwicklung energieoptimierter Fördergurte - Ein Gemeinschaftsprojekt der Conveyor Belt Group der ContiTech AG und der RWE Power AG. In: *World of Mining* (2008)
- [35] JEINSCH, T.: *Modellgestützte Verfahren zur Fehlerüberwachung und fehlerdiagnose sowie deren Anwendung in einem Informationssystem für Gurtförderanlagen.* VDI Verlag GmbH Düsseldorf, 2002
- [36] JEINSCH, T. ; DING, S. X. ; ENGEL, P. ; JAHN, W.: Verfahren zur Offline- und/oder Onlinequalitätsverfolgung von Fördergut über Gurtförderanlagen. In: *Amt. Kennz. 199 26 598.4* (1999)
- [37] JEINSCH, T. ; ENGEL, P. ; DING, S. X. ; SCHOCH, D. ; NEUMANN, H.: Statistische Verfahren zur Fehlererkennung und -diagnose in der Automatisierungstechnik. In: *atp - Automatisierungstechnische Praxis, Oldenbourg Industrieverlag* (2010)
- [38] JOLLIFFE, I. T.: *Principal Component Analysis.* New York, Berlin : Springer-Verlag, 1986
- [39] KESSLER, W.: *Multivariate Datenanalyse für die Pharma-, Bio- und Prozessanalytik.* Weinheim : WILEY-VCH Verlag, 2007
- [40] KING, G. ; BAUER, C. ; BERNARD, E. ; EBERSOLE, S.: *Hibernate Relational Persistence for Java and .NET Documentation.* <http://www.hibernate.org/docs>, 2011. – [Online accessed 2011-02-26]
- [41] KRUCHTEN, P.: Architectural Blueprints - The 4-1 View Model of Architecture. In: *IEEE Software 11/1995* (1995), Nr. 12(6), S. 42–50
- [42] KU, W. ; STOERER, R. H. ; GEORGAKIS, C.: Disturbance detection and isolation by dynamic principal component analysis. In: *Chemometrics and Intelligent Laboratory Systems* (1995)

- [43] LEE, J. M. ; YOO, C. K. ; CHOI, S. W. ; VANROLLEGHEM, P. A. ; LEE, I. B.: Nonlinear process monitoring using kernel principal component analysis. In: *Chemical Engineering Science*, 59 (1): 223-234 (2004)
- [44] LI, G. ; QIN, S. J. ; JI, Y. ; ZHOU, D.: Reconstruction based fault diagnosis for continous processes. In: *Control Engineering Practice* (2010)
- [45] LI, W. ; YUE, H. ; VALLE-CERVANTES, S. ; QIN, S. J.: Rekursive PCA for adaptive process monitoring. In: *Journal of Process Control* 10 (2000), S. 471–486
- [46] LITTLE, R. ; RUBIN, D.: *Statistical Analysis with missing Data*. John Wiley and Sons, 1976
- [47] MINA, J. ; VERDE, C.: Fault Detection for Large Scale Systems Using Dynamic Principal Components Analysis with Adaptation. In: *International Journal of Computers, Communications and Control* (2007)
- [48] NAIK, A. S. ; YIN, S. ; DING, S. X. ; JEINSCH, T.: Recursive identification algorithm for parity space based fault detection systems. In: *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain* (2009)
- [49] NORMUNG E.V., Deutsches I.: *Stetigförderer, Gurtförderer für Schüttgüter, Grundlagen für die Berechnung und Auslegung*, 1982
- [50] OWCZAREK, A.: *Recherche und Evaluierung geeigneter numerischer Verfahren und Frameworks zur statistischen Auswertung großer Datenmengen auf Basis von Java*, Hochschule Lausitz, Diplomarbeit, 2010
- [51] QIN, S. J.: Statistical process monitoring: basics and beyond. In: *Journal of Chemometrics* (2003)
- [52] QIN, S. J.: Data-driven Fault Detection and Diagnosis for Complex Industrial Processes. In: *Proc. of 7th IFAC SAFEPROCESS Symp., Barcelona* (2009)
- [53] QIN, S. J. ; LI, W.: Detection and Identification of Faulty Sensors Dynamic Processes. In: *AIChE Journal* (2001)
- [54] QINGSONG, Y.: *Model-based and Data Driven Fault Diagnosis Methods with Application to Process Monitoring*, Case Western Reserve University, Diss., 2004
- [55] ROSS, S. M.: *Statistik für Ingenieure und Naturwissenschaftler*. Spektrum Akademischer Verlag, 2006
- [56] ROSS, S. M.: *Statistik für Ingenieure und Naturwissenschaftler*. 3. Auflage. Heidelberg : Spektrum Akademischer Verlag, 2006
- [57] ROUSSEEUW, P.J. ; CROUX, C.: Alternatives to median absolute deviation. In: *Journal of the American Statistical Association* Vol. 88 (1993), Nr. 1273-1283

- [58] RUSSEL, E. L. ; CHIANG, L. H. ; BRAATZ, R. D.: *Data-Driven Techniques for Fault Detection and Diagnosis in Chemical Processes*. 1. Auflage. London Berlin Heidelberg : Springer-Verlag, 2000
- [59] RUSSEL, E. L. ; CHIANG, L. H. ; BRAATZ, R. D.: Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. In: *Chemometrics and Intelligent Laboratory Systems* 51 (2000), S. 81?93
- [60] SCHÖLKOPF, B. ; SMOLA, A. ; MÜLLER, K.: Nonlinear component analysis as a kernel eigenvalue problem. In: *Neural Computation* 10 (5) (1998)
- [61] SCHOCH, D. ; ENGEL, P. ; STARGALA, T. ; DING, S. X. ; JEINSCH, T.: Untersuchung datenbasierter Verfahren zur Prozessüberwachung und Fehlerdiagnose von komplexen, mechatronischen Systemen. In: *5. ASIM-Workshop Modellierung, Regelung und Simulation in Automotive und Prozessautomation* (2011)
- [62] SCHOMAKER, M.: *Neuere Ansätze für Kriterien zur Modellselektion bei Regressionsmodellen unter Berücksichtigung der Problematik fehlender Daten*, Institut für Statistik der Universität München, Diplomarbeit, 2006
- [63] SHLENS, J.: A Tutorial on Principal Component Analysis. (2005)
- [64] SOURCEFORGE: *Checkstyle 5.3*. <http://checkstyle.sourceforge.net>, 2010-10-19. – [Online accessed 2011-01-14]
- [65] SPITZLSPERGER, G. ; SCHMIDT, C. ; ERNST, G. ; STRASSER, H. ; SPEIL, M.: Fault Detection for a Via Etch Process Using Adaptive Multivariate Methods. In: *IEEE Transactions on Semiconductor Manufacturing* (2005)
- [66] STARKE, G.: *Effektive Software-Architekturen*. Hanser Verlag, 2005
- [67] STARKE, G. ; HRUSCHKA, P.: *Software-Architektur Kompakt: angemessen und zielorientiert*. Spektrum Akademischer Verlag; Auflage: 1, 2009
- [68] UNIVERSITY OF MARYLAND: *FindBugs Bug Descriptions*. <http://findbugs.sourceforge.net/bugDescriptions.html>, 2009. – [Online accessed 2011-01-14]
- [69] VAPNIK, V. N.: *The Nature of Statistical Learning Theory*. New York, Berlin : Springer-Verlag, 1995
- [70] VENKATASUBRAMANIAN, V. ; RENGASWAMY, R. ; KAVURI, S.: A review of process fault detection and diagnosis Part II: Qualitative models and search strategies. In: *Computers and Chemical Engineering* 56 (2003), S. 313–326
- [71] VENKATASUBRAMANIAN, V. ; RENGASWAMY, R. ; KAVURI, S. ; YIN, K.: A review of process fault detection and diagnosis Part I: Quantitative model-based methods. In: *Computers and Chemical Engineering* 56 (2003), S. 293–311

- [72] VENKATASUBRAMANIAN, V. ; RENGASWAMY, R. ; KAVURI, S. ; YIN, K.: A review of process fault detection and diagnosis Part III: Process history based methods. In: *Computers and Chemical Engineering* 56 (2003), S. 327–346
- [73] WEI, X. ; VERHAEGEN, M. ; ENGELN, T. van: Sensor fault detection and isolation for wind turbines based on subspace identification and Kalman filter techniques. In: *International journal of adaptive control and signal processing* (2009)

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe.

Senftenberg, 30. April 2012